

Faculdade de Engenharia da Universidade do Porto



# **Melhoramento da interface local de um Robô de Limpeza através da utilização do comando da consola Wii**

Daniel Tiago de Jesus Coutinho

Dissertação realizada no âmbito do Mestrado Integrado em Engenharia  
Electrotécnica e de Computadores Major Automação

Orientador: Prof. Dr. Armando Jorge Miranda de Sousa

Co-orientador: Prof. Dr. Luís Paulo Gonçalves dos Reis

Julho 2008



A Dissertação intitulada

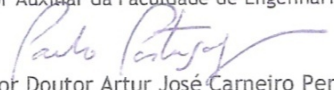
**“Melhoramento da Interface local de um Robô de Serviço através de utilização de comando da consola WII”**

foi aprovada em provas realizadas em 16/Julho/2008

o júri


Presidente

Professor Doutor Paulo José Lopes Machado Portugal  
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

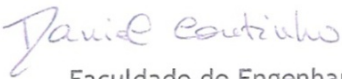
  
Professor Doutor Artur José Carneiro Pereira  
Professora Auxiliar da Universidade de Aveiro

  
Professor Doutor Armando Jorge Miranda de Sousa  
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

  
Professor Doutor Luis Paulo Gonçalves dos Reis  
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

  
O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - Daniel Tiago de Jesus Coutinho



Faculdade de Engenharia da Universidade do Porto



# Resumo

Com o aumento das capacidades de processamento e análise, os sistemas robóticos tornam-se cada vez mais capazes de desempenhar tarefas específicas de uma forma eficaz e eficiente, o que permite a sua integração num sistema de trabalho de cooperação com o ser humano. No entanto, para que esta coexistência se dê de uma forma sã e eficiente torna-se necessário desenvolver formas de interacção entre as duas partes.

Neste documento é apresentada a análise e implementação de uma interface homem-máquina local baseada no controlador remoto da consola Nintendo Wii, o Wiimote, sendo utilizado um robô de limpeza como plataforma de testes. Foram exploradas as capacidades que tornam este controlador remoto único, o que resultou na criação de três modos de funcionamento baseados nos seus diferentes componentes internos. O primeiro modo de funcionamento implementado foi o funcionamento como volante. Este modo tira partido das capacidades de detecção de acelerações do Wiimote para simular um volante e os pedais de aceleração e travagem de um automóvel. O controlador deste modo de funcionamento baseia-se em lógica difusa e é do tipo Mamdani. O segundo modo implementado foi a monitorização de infra-vermelhos. Neste modo recorre-se ao receptor de infra-vermelhos do Wiimote para captar fontes luminosas desse tipo e identifica a distância e orientação a que se encontram, podendo esta informação ser utilizada num algoritmo de auto-localização implementado no robô. Ainda neste modo foi criado o submodo follow-the-leader que utiliza esta informação para manter o Wiimote a uma distância e orientação fixa dessas fontes infra-vermelhas, havendo necessidade de este estar acoplado a um elemento móvel controlável, como o robô. O controlador utilizado neste submodo é também difuso do tipo Mamdani. Finalmente foi implementado o modo de identificação de movimentos que detecta e identifica os movimentos do utilizador. Neste modo tira-se novamente partido das capacidades de detecção de acelerações do comando para identificar características das acelerações lidas que permitem diferenciar movimentos lineares de movimentos circulares, o sentido do movimento e a direcção em que ele é feito. A técnica de controlo utilizada no controlador deste modo é raciocínio baseado em casos.

Através dos resultados obtidos para cada um dos modos de funcionamento mencionados, prova-se que o Wiimote é uma forma barata e fácil de implementar uma interface homem-máquina intuitiva e multi-operacional.



# Abstract

The increasing computing and analysis capacities improve the robotic systems abilities to do specific tasks more efficiently, which allow them to be integrated in human-robot cooperative working systems. In order to make this coexistence healthy and efficient, it is necessary to develop ways of interaction between them.

This document presents the analysis and implementation of a local human-machine interface based on the Nintendo Wii's remote controller, the Wiimote, using a cleaning robot as a test platform. The unique capacities of this remote controller were studied in order to create a three working modes based on its set of features. The first mode to be implemented was a steering wheel in which the remote controller is used to simulate the steering wheel and the acceleration and brake pedals of a car. This is done using the remote's capacity to read a tridimensional acceleration. To control this mode it was used a Mamdani fuzzy logic controller. The second working mode to be implemented was an infra-red monitor. This mode uses the Wiimote's infra-red sensor to detect light sources of this type and measures their distance and orientation in relation to the Wiimote, which can be used in an auto-location algorithm implemented in the robot. This information can be used in the follow-the-leader submode that ensures that the remote is kept at a fixed distance and orientation from the light sources, if it is attached to a controllable mobile element of the system, like the cleaning robot. This submode also uses a Mamdani fuzzy controller. Lastly, it was implemented a movement identification mode where the user's movements are detected and identified. This mode also uses the remote's capacity to read a tridimensional acceleration, in order to identify the acceleration's characteristics that differentiate linear and circular movements, the movement's orientation and its direction. The control technique used in this mode's controller was case based reasoning.

Analyzing the results of all the implemented working modes, it is shown that the Wiimote is an inexpensive and simple way to do a multi-operational intuitive human-machine interface.





# Agradecimentos

Gostaria de agradecer:

Aos meus pais e ao resto da minha família  
pelo apoio

Aos meus orientadores Armando Sousa e Luís Paulo Reis  
por me obrigarem a melhorar

A todos os meus colegas e amigos da FEUP que  
sempre estiveram dispostos a ajudar

A todos os que me ajudaram ao longo desta caminhada

Muito Obrigado



# Índice

|                                      |            |
|--------------------------------------|------------|
| <b>Resumo .....</b>                  | <b>v</b>   |
| <b>Abstract .....</b>                | <b>vii</b> |
| <b>Agradecimentos .....</b>          | <b>ix</b>  |
| <b>Índice .....</b>                  | <b>xi</b>  |
| <b>Lista de Figuras .....</b>        | <b>xv</b>  |
| <b>Lista de Tabelas .....</b>        | <b>xix</b> |
| <b>Abreviaturas e Símbolos .....</b> | <b>xxi</b> |

## Capítulo 1

|   |          |
|---|----------|
| <b>Introdução e Objectivos .....</b>          | <b>1</b> |
| 1.1 Introdução .....                          | 1        |
| 1.2 Estado da Arte .....                      | 2        |
| 1.2.1 Comunicações sem fios .....             | 2        |
| 1.2.2 Utilização do Wiimote .....             | 3        |
| 1.2.3 Estado Inicial do Robô de Limpeza ..... | 4        |
| 1.2.4 Interfaces Homem-Máquina .....          | 5        |
| 1.3 Objectivos .....                          | 5        |
| 1.4 Estrutura do Documento .....              | 6        |

## Capítulo 2

|   |          |
|---|----------|
| <b>Controlo Inteligente .....</b>                 | <b>7</b> |
| 2.1 Controladores Baseados em Lógica Difusa ..... | 7        |
| 2.2 Redes Neurais e Neuro-Difusas .....           | 8        |
| 2.3 Raciocínio Baseado em Casos .....             | 10       |
| 2.4 Conclusões .....                              | 11       |

### Capítulo 3

|                                       |    |
|---------------------------------------|----|
| <b>Estudo do Wiimote</b> .....        | 13 |
| 3.1 Acelerómetro Tridimensional ..... | 13 |
| 3.2 Receptor de Infra-Vermelhos ..... | 15 |
| 3.3 Restantes Constituintes .....     | 16 |
| 3.4 Conclusões .....                  | 16 |

### Capítulo 4

|   |    |
|---|----|
| <b>Solução Proposta</b> .....           | 17 |
| 4.1 Introdução .....                    | 17 |
| 4.2 Estrutura da Solução Proposta ..... | 17 |
| 4.3 Conclusões .....                    | 18 |

### Capítulo 5

|   |    |
|---|----|
| <b>Software Desenvolvido</b> .....                          | 19 |
| 5.1 Introdução .....  | 19 |
| 5.2 Aplicação de Interface .....                            | 19 |
| 5.2.1 Utilização da biblioteca libwiimote .....             | 19 |
| 5.2.2 Estabelecimento da comunicação entre aplicações ..... | 20 |
| 5.2.3 Descrição de Funcionamento .....                      | 21 |
| 5.3 Aplicação de Controlo .....                             | 22 |
| 5.3.1 Funcionamento como Volante.....                       | 23 |
| 5.3.2 Monitorização de Infra-Vermelhos.....                 | 28 |
| 5.3.3 Identificação de Movimentos .....                     | 32 |
| 5.4 Actuação sobre o Robô.....                              | 37 |
| 5.5 Conclusão .....   | 37 |

### Capítulo 6

|  |    |
|--|----|
| <b>Análise de Resultados</b> .....         | 39 |
| 6.1 Funcionamento como Volante .....       | 39 |
| 6.2 Monitorização de Infra-Vermelhos ..... | 39 |
| 6.3 Identificação de Movimentos .....      | 40 |
| 6.4 Conclusões .....                       | 43 |

**Capítulo 7**

|  |           |
|--|-----------|
| <b>Conclusões e Trabalho Futuro.....</b> | <b>45</b> |
| 7.1 Satisfação de Objectivos .....       | 45        |
| 7.2 Trabalho Futuro.....                 | 46        |

|  |           |
|--|-----------|
| <b>Referências Bibliográficas.....</b> | <b>49</b> |
|--|-----------|

**Anexo 1**

|                                    |           |
|------------------------------------|-----------|
| <b>Movimentos Analisados .....</b> | <b>51</b> |
|------------------------------------|-----------|

**Anexo 2**

|                                    |           |
|------------------------------------|-----------|
| <b>Tabelas de Resultados .....</b> | <b>57</b> |
|------------------------------------|-----------|



# Lista de Figuras

|   |    |
|---|----|
| <b>Figura 1.1</b> - Wiimote .....   | 2  |
| <b>Figura 1.2</b> – Robô de limpeza utilizado: CleanRob [11] .....  | 4  |
| <b>Figura 1.3</b> - Esquema de um robô de tracção diferencial .....   | 4  |
| <b>Figura 2.1</b> - Estrutura de um controlador difuso (adaptado de [21]).....  | 8  |
| <b>Figura 2.2</b> - Estrutura de um neurónio .....  | 9  |
| <b>Figura 2.3</b> - Ciclo CBR definido por Agnar Aamodt e Enric Plaza em [20] .....   | 10 |
| <b>Figura 3.1</b> - Sistema de eixos do Wiimote.....  | 13 |
| <b>Figura 3.2</b> – Exemplo típico das acelerações lidas pelo acelerómetro do eixo x quando o movimento feito é: a) linear para a esquerda b) linear para a direita ..... | 14 |
| <b>Figura 3.3</b> – Exemplo das acelerações lidas nos três eixos do acelerómetro num movimento linear para a esquerda: a) eixo x b) eixo y c) eixo z.....                 | 14 |
| <b>Figura 3.4</b> – Exemplo das acelerações lidas quando o é feito um movimento circular no plano xz no sentido horário: a) eixo x b) eixo y c) eixo z .....              | 14 |
| <b>Figura 3.5</b> - Representação da faixa de não detecção do sensor infra-vermelho .....   | 15 |
| <b>Figura 4.1</b> - Diagrama de funcionamento geral do software desenvolvido .....  | 17 |
| <b>Figura 5.1</b> - Estrutura da trama a enviar pela Aplicação de Interface .....   | 20 |
| <b>Figura 5.2</b> - Estrutura da trama a receber pela Aplicação de Interface .....  | 21 |
| <b>Figura 5.3</b> - Diagrama de fluxo da Aplicação de Interface.....  | 22 |
| <b>Figura 5.4</b> - Posição de actuação nula .....  | 23 |
| <b>Figura 5.5</b> - Repartição da gravidade quando o Wiimote está inclinado horizontalmente .....   | 23 |
| <b>Figura 5.6</b> - Repartição da gravidade quando o Wiimote está inclinado verticalmente .....   | 24 |
| <b>Figura 5.7</b> - Cálculo do valor de pertença de uma variável de valor x num conjunto difuso ....  | 26 |
| <b>Figura 5.8</b> - Representação gráfica das operações realizadas sobre os singletons que definem os conjuntos difusos das saídas .....                                  | 27 |
| <b>Figura 5.9</b> - Diagrama de fluxo do funcionamento como volante .....   | 28 |

|  |    |
|--|----|
| <b>Figura 5.10</b> - Esquema das distâncias utilizadas neste modo de funcionamento.....  | 29 |
| <b>Figura 5.11</b> – Esquema planar das distâncias dos pontos detectados ao centro (em pixel) e dos ângulos desses pontos em relação ao ponto do Wiimote.....                                  | 29 |
| <b>Figura 5.12</b> - Alteração dos conjuntos difusos da saída $r_1$ no modo follow-the-leader.....   | 31 |
| <b>Figura 5.13</b> - Diagrama de fluxo do modo de monitorização de infra-vermelhos .....   | 31 |
| <b>Figura 5.14</b> - Diagrama de fluxo do modo de identificação de movimentos.....   | 33 |
| <b>Figura 5.15</b> – Comparação da aceleração lida (a traço contínuo) com a aceleração obtida após o processamento do algoritmo de eliminação de máximos e mínimos locais (a tracejado), ..... | 35 |
| <b>Figura 5.16</b> - Representação de $\alpha$ e $\beta$ no espaço tridimensional.....   | 36 |
| <b>Figura 6.1</b> - Variação do módulo do erro das medições em relação à distância real .....  | 40 |
| <b>Figura 6.2</b> - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos ao longo do eixo x do Wiimote.....   | 41 |
| <b>Figura 6.3</b> - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos ao longo do eixo y do Wiimote.....   | 41 |
| <b>Figura 6.4</b> - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos ao longo do eixo z do Wiimote .....  | 41 |
| <b>Figura 6.5</b> - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos feitos no plano xz do Wiimote .....  | 42 |
| <b>Figura A1.1</b> – Aceleração ao longo do eixo x para um movimento: a) linear para a esquerda, b) linear para a direita.....   | 51 |
| <b>Figura A1.2</b> – Aceleração ao longo do eixo y para um movimento: a) linear para a esquerda, b) linear para a direita.....   | 51 |
| <b>Figura A1.3</b> – Acelerações ao longo do eixo z para um movimento: a) linear para a esquerda, b) linear para a direita.....  | 52 |
| <b>Figura A1.4</b> – Acelerações ao longo do eixo x para um movimento: a) linear para a frente, b) linear para a retaguarda .....  | 52 |
| <b>Figura A1.5</b> – Acelerações ao longo do eixo y para um movimento: a) linear para a frente, b) linear para a retaguarda .....  | 52 |
| <b>Figura A1.6</b> – Acelerações ao longo do eixo z para um movimento: a) linear para a frente, b) linear para a retaguarda .....  | 52 |
| <b>Figura A1.7</b> – Acelerações ao longo do eixo x para um movimento: a) linear para cima, b) linear para baixo.....  | 52 |
| <b>Figura A1.8</b> – Acelerações ao longo do eixo y para um movimento: a) linear para cima, b) linear para baixo.....  | 53 |
| <b>Figura A1.9</b> – Acelerações ao longo do eixo z para um movimento: a) linear para cima, b) linear para baixo.....  | 53 |
| <b>Figura A1.10</b> - Aceleração ao longo do eixo x num movimento linear repetitivo ao longo deste eixo.....   | 53 |



|  |    |
|--|----|
| <b>Figura A1.11</b> - Aceleração ao longo do eixo y num movimento linear repetitivo ao longo do eixo x .....                                 | 53 |
| <b>Figura A1.12</b> - Aceleração ao longo do eixo z num movimento linear repetitivo ao longo do eixo x .....                                 | 53 |
| <b>Figura A1.13</b> - Aceleração ao longo do eixo x num movimento linear repetitivo ao longo do eixo y .....                                 | 54 |
| <b>Figura A1.14</b> - Aceleração ao longo do eixo y num movimento linear repetitivo ao longo deste eixo .....                                | 54 |
| <b>Figura A1.15</b> - Aceleração ao longo do eixo z num movimento linear repetitivo ao longo do eixo y .....                                 | 54 |
| <b>Figura A1.16</b> - Aceleração ao longo do eixo x num movimento linear repetitivo ao longo do eixo z .....                                 | 54 |
| <b>Figura A1.17</b> - Aceleração ao longo do eixo y num movimento linear repetitivo ao longo do eixo z .....                                 | 54 |
| <b>Figura A1.18</b> - Aceleração ao longo do eixo z num movimento linear repetitivo ao longo deste eixo .....                                | 55 |
| <b>Figura A1.19</b> – Acelerações ao longo do eixo x para um movimento circular no plano xz no sentido: a) horário, b) anti-horário .....    | 55 |
| <b>Figura A1.20</b> - Acelerações ao longo do eixo y para um movimento circular no plano xz no sentido: a) horário, b) anti-horário .....    | 55 |
| <b>Figura A1.21</b> - Acelerações ao longo do eixo z para um movimento circular no plano xz no sentido: a) horário, b) anti-horário .....    | 55 |
| <b>Figura A1.22</b> - Acelerações ao longo do eixo x para um movimento rectangular no plano xz no sentido: a) horário, b) anti-horário ..... | 55 |
| <b>Figura A1.23</b> - Acelerações ao longo do eixo x para um movimento rectangular no plano xz no sentido: a) horário, b) anti-horário ..... | 56 |
| <b>Figura A1.24</b> - Acelerações ao longo do eixo x para um movimento rectangular no plano xz no sentido: a) horário, b) anti-horário ..... | 56 |



# Lista de Tabelas

|  |    |
|--|----|
| <b>Tabela 5.1</b> - Valores das acelerações lidas nos nove pontos de controlo definidos .....      | 24 |
| <b>Tabela 5.2</b> - Base de regras utilizada.....  | 25 |
| <b>Tabela 5.3</b> - Expressões de cálculo do nível de disparo das regras .....                     | 26 |
| <b>Tabela A2.1</b> - Valores médios das distâncias medidas com o receptor de Infra-Vermelhos ..... | 57 |
| <b>Tabela A2.2</b> - Resultados dos movimentos ao longo do eixo x .....                            | 58 |
| <b>Tabela A2.3</b> - Resultados dos movimentos ao longo do eixo y .....                            | 58 |
| <b>Tabela A2.4</b> - Resultados dos movimentos segundo o eixo z.....                               | 59 |
| <b>Tabela A2.5</b> - Resultados dos movimentos feitos ao longo de vários eixos .....               | 59 |



# Abreviaturas e Símbolos

## Lista de abreviaturas

|     |                                       |
|-----|---------------------------------------|
| CBR | <i>Case Based Reasoning;</i>          |
| GPS | <i>Global Positioning System;</i>     |
| LED | <i>Light Emitting Diode;</i>          |
| PC  | <i>Personal Computer;</i>             |
| PWM | <i>Pulse Width Modulation;</i>        |
| SNR | <i>Signal-to-Noise Ratio;</i>         |
| TCP | <i>Transmission Control Protocol;</i> |
| UDP | <i>User Datagram Protocol;</i>        |

## Lista de símbolos

|                |   |
|----------------|---|
| $v$            | Velocidade linear;  |
| $\omega$       | Velocidade angular;   |
| $v_{nom}$      | Velocidade linear nominal do robô;  |
| $\omega_{nom}$ | Velocidade angular nominal do robô;   |
| $v_{out}$      | Valor de actuação sobre a velocidade linear obtido no sistema de inferência difuso;   |
| $w_{out}$      | Valor de actuação sobre a velocidade angular obtido no sistema de inferência difuso;  |
| $\phi$         | Ângulo da abertura horizontal do receptor de infra-vermelhos do Wiimote;  |
| $\gamma$       | Ângulo da abertura vertical do receptor de infra-vermelhos do Wiimote;  |
| $d$            | Distância entre os dois focos infra-vermelhos expressa em centímetros;  |
| $h$            | Distância máxima a que dois pontos luminosos com espaçamento $d$ são detectados pelo Wiimote quando o seu ponto médio é coincidente com a linha vertical média da resolução do receptor de infra-vermelhos ( $x=512$ ); |

|            |  |
|------------|--|
| $\alpha_k$ | Ângulo entre a aceleração parcial lida no eixo x do Wiimote e a projecção da aceleração do mesmo sobre o plano xy no instante de amostragem k; |
| $\beta_k$  | Ângulo entre a aceleração do Wiimote e a sua projecção sobre o plano xy no instante de amostragem k;   |
| $accx_k$   | Aceleração lida pelo acelerómetro orientado segundo o eixo x do Wiimote no instante de amostragem k;   |
| $accy_k$   | Aceleração lida pelo acelerómetro orientado segundo o eixo y do Wiimote no instante de amostragem k;   |
| $accz_k$   | Aceleração lida pelo acelerómetro orientado segundo o eixo z do Wiimote no instante de amostragem k;   |
| $acc_k$    | Módulo da aceleração do Wiimote no instante de amostragem k;   |

# Capítulo 1

## Introdução e Objectivos

Neste capítulo serão apresentadas as considerações básicas relativas à elaboração deste trabalho, o estado de desenvolvimento da arte aquando do início do mesmo e os objectivos que se pretendem atingir.

### 1.1 Introdução

A 8 de Dezembro de 2006 foi lançada na Europa a *Wii*, a quinta consola de jogos da Nintendo [1]. Apesar desta consola ter como mercado alvo o mercado de entretenimento infantil/familiar tornou-se num sucesso de vendas junto do grande público devido ao seu inovador método de controlo. Ao invés de utilizar um comando tradicional onde o controlo é feito através de uma combinação de botões, a *Wii* dispõe de um comando sem fios capaz de interpretar os movimentos dos jogadores, o *Wiimote* (abreviatura de *Wii Remote*).

Este método de controlo inovador despertou a curiosidade de diversas entidades que pretendiam aproveitar-se da sua electrónica interna já devidamente condicionada e tratada e da capacidade de comunicação *wireless* oferecida pelo *Wiimote* para desenvolver sistemas onde este comando pudesse ser utilizado. Da electrónica mencionada destaca-se um receptor de infra-vermelhos com capacidade de *tracking* de quatro fontes infra-vermelhas, no máximo, um acelerómetro tridimensional com 8bit de resolução em cada eixo, quatro *LEDs* indicadores do estado da bateria e a capacidade de comunicação *Bluetooth*. Foi aliás graças a esta forma de comunicação e à análise das tramas enviadas pelo comando que a comunidade conseguiu descobrir de que forma os diversos elementos internos eram controláveis, possibilitando o uso de um objecto patenteado e cujo *software* não estava acessível para as mais diversas aplicações. Actualmente estão identificados mais de uma dezena de modos de comunicação que permitem controlar, isto é ler e escrever, os estados dos botões, acelerómetros, infra-vermelhos e outros periféricos que possam ser ligados ao *Wiimote*.



Figura 1.1 - Wiimote

## 1.2 Estado da Arte

### 1.2.1 Comunicações sem fios

Ano após ano o nível de conectividade global aumenta e na última década um grande parte desse aumento deve-se à disponibilização de diversas tecnologias *wireless* ao nível do grande público. Exemplo disso é a generalização dos telemóveis na vida quotidiana ou o aparecimento de serviços como televisão por satélite ou o *Global Positioning System* (GPS). Ao nível da indústria este tipo de tecnologia passou a fazer parte integrante dos sistemas de comunicação devido às várias vantagens que lhe estão associadas, tais como a redução da cablagem existente e a consequente redução de custos e o facto de serem o tipo de tecnologias de comunicação que mais facilitam a construção e manutenção de sistemas modulares.

Apesar da atenção de que são alvo as comunicações *wireless* actualmente, estas já são utilizadas há bastante tempo em sistemas de controlo remoto. A utilização deste tipo de comunicação neste tipo de sistemas representa uma enorme vantagem já que oferecem uma enorme liberdade ao utilizador. Dois exemplos que demonstram esta vantagem são os comandos remotos de televisores e de garagens. Do ponto de vista tecnológico este tipo de sistemas comunica maioritariamente através de infra-vermelhos, no entanto este tipo de tecnologia é bastante susceptível a interferências e ruídos e o raio de acção também é pequeno. Recentemente a aposta mais forte ao nível das comunicações *wireless* de curta distância é o *Bluetooth*. Comparando estes dois meios de comunicação chega-se à conclusão que o *Bluetooth* permite aumentar o raio de acção de alguns metros para algumas dezenas de metros, opera a frequências muito superiores o que lhe confere maior robustez contra ruídos e sinais parasitas e a sua largura de banda é também superior. Outra tecnologia de comunicações *wireless* já implementada é o *Wi-Fi* que em termos técnicos é similar ao *Bluetooth* mas oferece ligações mais fortes através de um maior nível de configuração. Juntas estas três tecnologias permitem satisfazer todas as necessidades de comunicações *wireless* de curta distância existentes: os infra-vermelhos são baratos sendo frequentemente encontrados em sistemas de controlo remoto; o *Bluetooth* é utilizado quando se pretende maior fiabilidade mas a segurança da comunicação não é crítica; as redes *Wi-Fi* sobrepõem-se ao *Bluetooth* ao nível da segurança da rede tendo isso como custo a necessidade de um maior volume de configurações da comunicação.



### 1.2.2 Utilização do Wiimote

O lançamento da Nintendo *Wii* no Natal de 2006 marcou uma revolução ao nível dos sistemas de entretenimentos domésticos devido ao seu inovador sistema de controlo, o *Wiimote*, que permitia controlar as situações de jogo através dos movimentos do utilizador. Esta característica levou à exploração das suas capacidades pelas mais diversas entidades com vista a aplicá-lo a sistemas cujos objectivos seriam totalmente opostos ao seu objectivo original. Esta investigação resultou no levantamento dos seus diversos modos de operação e à sua divulgação de forma gratuita, tornando possível a utilização deste equipamento em qualquer sistema [2]. Actualmente existem diversas bibliotecas que criam a interface *Wiimote-PC*, ou seja, analisam as tramas enviadas pelo comando e disponibilizam a informação conforme o seu tipo, isto é, o estado dos botões, o valor lido pelo acelerómetro do eixo x, etc, o que facilita ainda mais a utilização deste equipamento. Estas bibliotecas podem ser *cross-platform* ou *single-platform*, existindo versões para Windows e Linux, e estão disponíveis em diversas linguagens, tais como C, Java, Delphi, entre outras [3].

O aparecimento destas bibliotecas permitiu que todos os utilizadores, com mais ou menos conhecimentos técnicos, desenvolvessem diversas aplicações com graus de complexidade variados. Entre estas aplicações destacam-se as desenvolvidas por Johnny Lee [4] e o conjunto de bibliotecas *CWiid* para Linux [5], apesar de nenhuma tirar partido das capacidades de detecção de movimentos do *Wiimote*. Esta capacidade está a ser explorada pelo exército norte-americano no manuseamento do PackBot [6][7].

Nas aplicações desenvolvidas por Johnny Lee o *Wiimote* encontra-se parado e é utilizada a capacidade de sensorização de fontes infra-vermelhas para calcular a posição relativa dessas fontes em relação ao *Wiimote*, sendo a actuação feita com base nas posições calculadas. São da sua autoria as aplicações '*Low-Cost Multi-point Interactive Whiteboards Using the Wiimote*' e '*Head Tracking for Desktop VR Displays using the Wii Remote*'. Na primeira o leitor de infra-vermelhos é utilizado para detectar a localização de um foco luminoso sobre uma superfície. Se nessa superfície for projectada a imagem de um monitor, o ponto infra-vermelho pode ser utilizado como o ponteiro de um rato, tornando a superfície outrora "branca" numa superfície interactiva, existindo ainda a possibilidade de adicionar novos pontos luminosos. Na segunda é calculada a posição, isto é, distância e ângulo, do ponto médio de duas fontes infra-vermelhas em relação ao *Wiimote* através do algoritmo de triangulação. Estas duas variáveis são depois utilizadas para criar um ambiente virtual baseado numa imagem plana onde os objectos dessa imagem podem ser vistos como maiores ou mais pequenos dependendo da distância e do ângulo lido. O conjunto de bibliotecas *CWiid* têm como objectivo o uso do *Wiimote* como rato/*joystick*. Nesta aplicação o *Wiimote* é o elemento móvel e são necessárias duas fontes infra-vermelhas fixas ao monitor para ser possível calcular o ponto para onde o *Wiimote* aponta, ou seja, o algoritmo de localização utilizado é a triangulação, tal como na aplicação de *desktop* virtual de Johnny Lee. Estes três exemplos provam a versatilidade do *Wiimote* já que ele pode ser um elemento móvel ou estático num sistema.

Existem outras aplicações desenvolvidas mas cuja documentação existente se resume a vídeos do seu funcionamento [8][9][10].

### 1.2.3 Estado Inicial do Robô de Limpeza

A figura 1.2 representa o sistema robótico utilizado, o CleanRob [11]. Aquando do início deste trabalho este robô era capaz de efectuar acções de movimentação simples e de se auto localizar através da fusão de informação dos seus diversos sensores: quatro *sharps* de 1,5 metros [12], seis sonares e uma câmara detectora de códigos de barras colocados nas paredes. Os algoritmos de auto-localização presentes no robô nesta fase eram o Filtro de Kalman Extendido [13] e o Filtro de Partículas [14] ainda em fase embrionária.

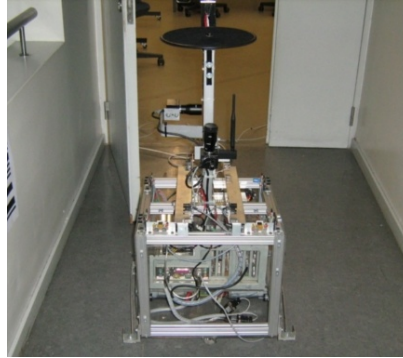


Figura 1.2 - Robô de limpeza utilizado: CleanRob [11]

O CleanRob é um robô de tracção diferencial e cuja cinemática pode ser analisada a partir do esquema da figura 1.3. O robô possui duas rodas sobre um eixo de tamanho  $b$ , sendo a velocidade de cada uma das rodas num instante de tempo dado por  $v_1$  e  $v_2$ . O valor das velocidades linear ( $v$ ) e angular ( $\omega$ ) do robô nesse instante é calculado através das expressões 1.1 e 1.2. Considerando que um controlador terá como saída estas duas velocidades, a actuação sobre os motores de cada uma das rodas é feita através do cálculo das velocidades  $v_1$  e  $v_2$  e o posterior envio para os controladores de cada um dos motores, no caso do CleanRob são utilizados controladores PID.

$$v = \frac{v_1 + v_2}{2} \quad (1.1)$$

$$\omega = \frac{v_1 - v_2}{b} \quad (1.2)$$

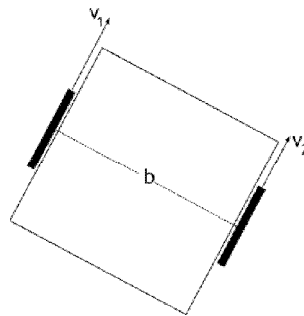


Figura 1.3 - Esquema de um robô de tracção diferencial

### 1.2.4 Interfaces Homem-Máquina

Com o aumento da quantidade e da qualidade dos robôs capazes de desempenhar tarefas humanas, torna-se necessário implementar interfaces homem-máquina o mais intuitivas possíveis de forma a garantir uma coexistência sã entre o homem e o seu semelhante robótico. A interface entre seres humanos é feita através das suas capacidades de comunicação por voz e por gestos, o que indica que as interfaces homem-máquina baseadas no reconhecimento destes dois tipos de dados serão as que oferecerão maior facilidade de operação a um utilizador [15][16][17]. Nestes trabalhos o reconhecimento gestual é feito através de câmaras colocadas nos robôs o que é um incentivo extra para este trabalho uma vez que o reconhecimento de gestos será feito através do *Wiimote*.

## 1.3 Objectivos

Neste trabalho pretendeu-se implementar um sistema de controlo remoto com base no *Wiimote* que fosse capaz de tirar partido das suas capacidades tendo como finalidade a sua aplicação num sistema robótico de limpeza. Para tal foram criados diversos objectivos listados de seguida:

1. Desenvolver um algoritmo responsável pela comunicação entre o *Wiimote* e o software de controlo;
2. Testar o funcionamento dos diferentes elementos do *Wiimote*, em particular:
  - 2.1. Fazer o mapeamento do bit representativo de cada botão na estrutura utilizada;
  - 2.2. Identificar características dos movimentos e o sistema de eixos do *Wiimote* utilizando os dados medidos pelo acelerómetro tridimensional;
  - 2.3. Analisar as características de funcionamento do receptor de infra-vermelhos tais como máxima distância a que é possível detectar uma fonte infra-vermelha e os ângulos de abertura horizontal e vertical;
3. Estudar os diferentes modos de funcionamento possíveis de serem implementados utilizando o *Wiimote* e o sistema robótico mencionado;
4. Analisar diferentes técnicas de controlo existentes e escolher a que melhor se adapta a cada modo de funcionamento;

Como objectivo secundário pretendeu-se que o sistema implementado fosse totalmente baseado em software grátis e que o seu funcionamento fosse independente do sistema operativo utilizado.

Dos objectivos apresentados foram todos cumpridos com excepção do objectivo secundário. O software onde esta interface de controlo local foi implementada é totalmente *opensource* mas não é *cross-platform* devido a uma dependência duma biblioteca utilizada.

## 1.4 Estrutura do Documento

No capítulo seguinte são apresentadas as três formas de controlo inteligente analisadas, com especial foco na sua forma de implementação.

No capítulo 3 são apresentadas as características dos constituintes do *Wiimote* que serão utilizados no desenvolvimento dos diversos modos de funcionamento implementados.

No capítulo 4 é apresentada a estrutura da solução proposta.

No capítulo 5 são apresentadas as aplicações desenvolvidas e as soluções de controlo que compõem a interface implementada.

Finalmente são apresentados os resultados obtidos, as conclusões retiradas durante e após a realização do trabalho e algumas propostas de melhoramentos futuros.

Em anexo encontram-se os gráficos das acelerações relativas aos movimentos analisados e as tabelas dos resultados obtidos.

## Capítulo 2

# Controlo Inteligente

Apesar de, nalguns casos, a sua fundamentação teórica ter sido formulada há dezenas ou centenas de anos, apenas recentemente o desenvolvimento tecnológico ao nível dos semicondutores permitiu que os processadores e memórias existentes fossem suficientemente rápidos e potentes de forma a tornar possível a implementação de técnicas de controlo inteligente. Estas técnicas de controlo divergem das técnicas de controlo tradicionais na abordagem que fazem do problema. Enquanto as técnicas clássicas abordam um problema de controlo através do estudo das equações e fundamentos matemáticos que definem e explicam esse problema, as técnicas inteligentes procuram uma abordagem mais próxima da abordagem feita pelo ser humano, isto é, sem recurso a equações ou a fundamentos teóricos elaborados, o que lhes confere um poder de controlo igual para sistemas lineares e não-lineares. Os controladores baseados nestas técnicas podem ser de vários tipos, sendo explorados três deles: controladores baseados em lógica difusa [18]; redes neuronais, podendo existir variantes como as redes neuro-difusas [19]; raciocínio baseado em casos (*Case Based Reasoning* – CBR) [20].

Neste capítulo são apresentados os fundamentos teóricos relativos à implementação de cada uma das técnicas de controlo mencionadas em qualquer sistema, sendo os seus pormenores matemáticos explicados em detalhe nas referências apresentadas acima.

### 2.1 Controladores Baseados em Lógica Difusa

Dos controladores baseados nas três técnicas de controlo inteligente analisadas, os controladores baseados em lógica difusa são os mais simples de implementar, actuando correctamente sobre sistemas lineares e não-lineares mesmo quando o sistema não é completamente conhecido, isto é, quando a base de regras criada não abrange todos os pontos de controlo onde o sistema se pode encontrar. Os controladores baseados em lógica difusa podem ser de dois tipos, Mamdani ou Takagi-Sugeno, e assentam o seu funcionamento em três fases, a *fuzificação* das variáveis de entrada, o mecanismo de inferência e a *desfuzificação* das variáveis de saída, e são constituídos por três grandes blocos, os conjuntos difusos das entradas, os conjuntos difusos das saídas e a base de regras, como está demonstrado na figura 2.1.

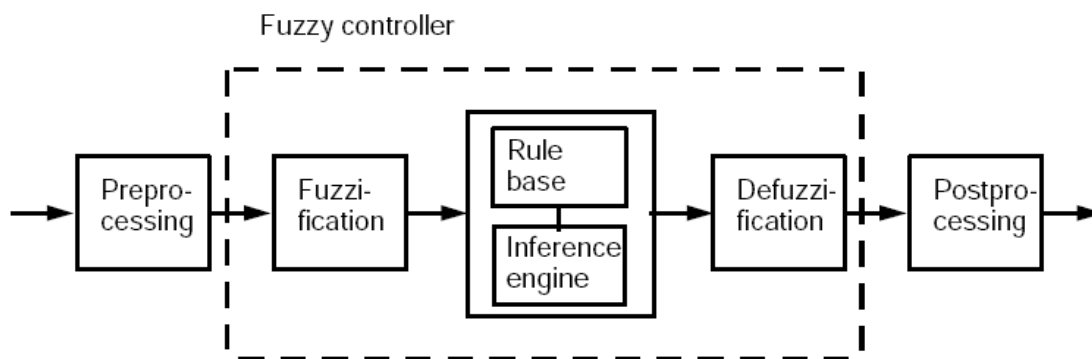


Figura 2.1 - Estrutura de um controlador difuso (adaptado de [21])

O algoritmo do controlador começa por transformar as variáveis de entrada em variáveis linguísticas, isto é, transforma as variáveis definidas por um valor algébrico em variáveis definidas por um conjunto difuso e pelo seu valor de pertença nesse conjunto. Os conjuntos podem ser triangulares, trapezoidais, impulsos (*singletons*), entre outros, podendo haver intersecções entre os conjuntos. O valor de pertença traduz a certeza que o sistema tem na localização dessa variável no conjunto em causa. Este processo de transformação da definição de uma variável designa-se por *fuzificação*. Com base nas regras existentes na base de regras previamente criada é processado o nível de disparo de cada uma dessas regras através do grau de satisfação da sua premissa. A premissa de uma regra é uma composição lógica onde são efectuadas operações de intercepção, reunião e complementaridade sobre os conjuntos difusos de entrada que nela estão representados, sendo estas operações representadas pelos operadores lógicos *AND*, *OR*, e *NOT*, respectivamente. Após analisar a premissa de uma regra é necessário analisar a sua consequência, sendo a sua forma o principal elemento que diferencia os controladores difusos do tipo Mamdani dos do tipo Takagi-Sugeno. Nos primeiros o consequente tem uma forma semelhante à premissa e o valor dos conjuntos difusos nele existentes é definido pelo nível de disparo dessa regra, sendo a saída calculada através da agregação dos vários consequentes e da *desfuzificação* através do centro de massa ou da média de máximos. Nos controladores de Takagi-Sugeno o consequente é uma função  $f$ , tipicamente linear, dos valores de pertença das variáveis existentes na premissa da regra sendo a saída  $u$  calculada através duma média pesada do valor obtido a partir dessas funções e do valor de disparo  $\eta$  de cada regra (equação 2.1).

$$u = \frac{\sum_k \eta_k \cdot f_k}{\sum_k \eta_k} \quad (2.1)$$

## 2.2 Redes Neurais e Neuro-Difusas

As redes neuronais são o tipo de controladores inteligentes que teoricamente mais se aproximam das capacidades e da forma de raciocínio do cérebro humano. São constituídas por camadas de neurónios e por ligações sinápticas entre os neurónios dessa camada e todos os neurónios da camada seguinte e anterior, podendo as camadas ser de três tipos: de entrada, se for a primeira camada da rede, de saída, se for a última, ou escondida, se for uma camada existente entre essas duas. As redes mais simples são constituídas por uma camada de cada tipo. Estas camadas são constituídas por neurónios que são os elementos responsáveis pelo

processamento da informação disponível e pela solução apresentada. Do ponto de vista da sua constituição um neurónio resume-se a uma função de transferência que responde a um vector de entradas e cuja saída é utilizada numa função de activação  $\varphi$ , onde é processada a saída do neurónio, podendo ainda existir um valor de polarização  $\theta$  que, do ponto de vista dos cálculos, funciona como uma entrada comum. O esquema de um neurónio pode ser visto na figura 2.2. As funções de activação mais comuns são o degrau por ser o mais simples e a função sigmóide por ser monótona e contínua.

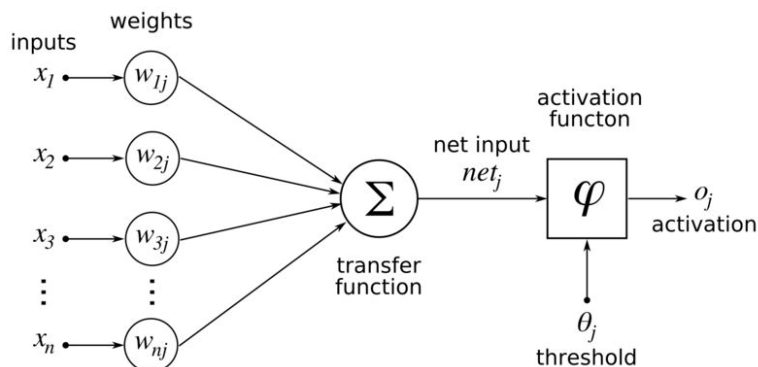


Figura 2.2 - Estrutura de um neurónio

Enquanto nos controladores difusos a inteligência estava associada à base de regras, nas redes neuronais esta está associada ao peso das ligações sinápticas entre cada neurónio, sendo necessário um algoritmo de treino destas redes que seja capaz de modular esses pesos de forma a obter a melhor qualidade de resposta possível. Dos algoritmos de treino existentes o mais frequentemente utilizado é o algoritmo de gradiente descendente. Neste algoritmo a variação do peso é calculado através da equação 2.2, onde  $E(\omega)$  é o somatório dos erro quadráticos entre o valor desejado à saída de um neurónio e o valor obtido e  $\rho$  é uma constante denominada de taxa de aprendizagem.

$$\Delta\omega = -\rho \cdot \nabla E(\omega) \quad (2.2)$$

As redes neuro-difusas são uma variação das redes neuronais onde se pretende tirar proveito das vantagens das duas abordagens. As redes neuronais proporcionam um controlo mais preciso enquanto os sistemas difusos são mais intuitivos e fáceis de configurar/implementar. A estrutura deste tipo de redes é bastante semelhante à estrutura das redes neuronais convencionais mas nas redes neuro-difusas a primeira camada é constituída pelos conjuntos difusos de entrada, isto é, é feita a *fuzificação* das entradas, o que faz com que os valores em trânsito dentro da rede sejam os valores de pertença das diversas variáveis de entrada. A última camada efectua a *desfuzificação* conforme o tipo de sistema de inferência escolhido, Mamdani ou Takagi-Sugeno. Este sistema de inferência é também utilizado nos neurónios para determinar a sua saída. O algoritmo de adaptação altera-se ligeiramente, continuando a ser possível modificar os valores dos pesos das ligações sinápticas sendo adicionada a possibilidade de alterar os valores que definem os conjuntos difusos existentes na rede, isto é, o valor da largura da base, o valor da posição do valor máximo desse conjunto e esse valor máximo.

## 2.3 Raciocínio Baseado em Casos

Os sistemas CBR funcionam como repositórios de problemas anteriormente resolvidos e as soluções encontradas para os resolver. Estes podem diferir bastante entre si no entanto todos seguem o esquema de inferência da figura 2.3 apresentado por Agnar Aamodt e Enric Plaza em [10].

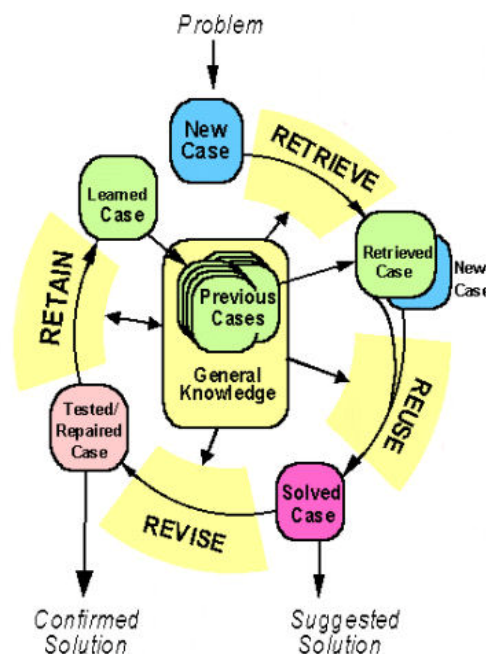


Figura 2.3 - Ciclo CBR definido por Agnar Aamodt e Enric Plaza em [20]

Este esquema define que um sistema CBR deve ser composto por quatro passos, a recolha (*Retrieve*), a reutilização (*Reuse*), a revisão (*Revise*) e a retenção (*Retain*), e por dois elementos fundamentais, a base de casos (*General Knowledge*) que é constituída por um conjunto de casos anteriormente resolvidos e as suas soluções e um novo caso para o qual é necessário encontrar uma solução. O primeiro passo do algoritmo CBR é recolher todas as informações úteis e necessárias do novo caso, sendo essa informação denominada de atributos. De seguida o novo caso é comparado com cada um dos casos guardados na base de casos de forma a encontrar o par “novo caso – caso guardado” que apresenta maior semelhança, ou menor diferença, sendo a solução do caso guardado adoptada como a solução do novo caso. A semelhança é medida através da comparação atributo a atributo entre os dois casos. No passo de revisão o algoritmo de treino verifica se a solução obtida no passo anterior é aplicável e caso não seja adapta essa solução de forma a torná-la aplicável. O passo final verifica se o caso que foi resolvido irá fornecer nova informação útil á base de casos, adicionando-o a essa base caso a resposta seja afirmativa. É este último passo que garante a capacidade de aprendizagem deste tipo de algoritmos. A complexidade destes algoritmos é variável e depende maioritariamente da dificuldade relacionada com o levantamento dos atributos que permitam diferenciar de uma forma consistente os diversos casos existentes. De forma análoga a qualidade das respostas obtidas depende da qualidade e quantidade de casos guardados na base de casos, o que permite retirar a conclusão que a solução correcta de um caso só é obtida se existirem casos desse tipo guardados na base de casos.



## 2.4 Conclusões

Neste capítulo foram apresentadas os fundamentos teóricos referentes à implementação prática das técnicas de controlo inteligente nele contido. O controlo difuso é bastante prático e fácil de utilizar em sistemas de baixa/média complexidade, isto é, com um número não muito elevado de variáveis de entrada e/ou saída, podendo estes ser lineares ou não-lineares. A decisão está associada á base de regras existentes que designam os pontos de controlo desejados para o sistema. As redes neuronais são os controladores inteligentes mais complexos mas que permitem respostas de alta qualidade. A sua capacidade de decisão está associada aos pesos das diversas ligações sinápticas entre os neurónios existentes nas diferentes camadas que a constituem. O raciocínio baseado em casos é uma técnica que tira partido do conhecimento existente de casos anteriores para catalogar e resolver novos casos que lhe são apresentados, de acordo com o caso conhecido mais semelhante ao caso novo. Por esta razão a sua capacidade de decisão está totalmente associada á qualidade e quantidade de casos conhecidos, isto é, resolvidos, existentes na sua base de casos.



## Capítulo 3

### Estudo do Wiimote

O *Wiimote* é constituído por diversos componentes, destacando-se aqueles que lhe garantem as capacidades mais visíveis ao utilizador como a sensorização de movimento e de posição e o facto de ser um elemento *wireless*. De seguida são expostas algumas características técnicas dos diversos constituintes do *Wiimote* determinadas experimentalmente.

#### 3.1 Acelerómetro Tridimensional

A capacidade de detecção de movimentos é garantida através da presença de um acelerómetro de três eixos. A gama de medição foi determinada através da análise das leituras feitas quando a única aceleração existente era a aceleração da gravidade, isto é, lendo o valor de 1G foi possível determinar a variações máxima e mínima possíveis de medir. Sendo a resolução das medições 8bit (valores pertencentes ao intervalo  $[0, 255]$ ), o valor médio de 1G medido foi 25 e tendo em conta que o valor médio da aceleração lida num eixo quando este não está sujeito a qualquer aceleração é 130, chega-se á conclusão que a gama de medição do acelerómetro tridimensional é  $\pm 5G$  em cada eixo. Outra característica retirada da análise das leituras do acelerómetro foi o sentido dos seus eixos, estando o sistema de eixos obtido representado na figura 3.1.

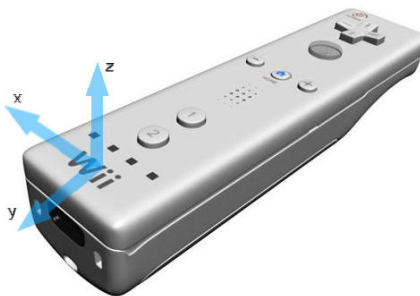


Figura 3.1 - Sistema de eixos do Wiimote

A última análise feita através dos acelerómetros foi a detecção de diferentes características dos movimentos, tendo particular interesse a análise de características das acelerações em cada

eixo que permitam diferenciar movimentos lineares de movimentos circulares, diferenciar em que eixo o movimento linear foi feito e em que sentido e diferenciar o sentido de um movimento circular. Tomando como exemplo os gráficos da aceleração ao longo do eixo x do *Wiimote* das figuras 3.2a e 3.2b, chega-se à conclusão que a aceleração ao longo do eixo em que um movimento linear é feito se assemelha a um período de um sinal sinusoidal, tal como esperado. Isto deve-se à inversão do sentido do vector aceleração aquando do momento de travagem e a posterior anulação da velocidade traduzida pelo anulamento entre as áreas delimitadas pelo valor nulo da aceleração e as acelerações antes e depois do instante de travagem. Considerando ainda estes dois movimentos lineares é fácil verificar que os sentidos são diferenciáveis através da ordem pela qual os máximos e mínimos são detectados. Se um movimento linear for feito no sentido positivo de um eixo o gráfico da aceleração nesse eixo será semelhante ao gráfico da aceleração da figura 3.2a, caso o movimento seja no sentido contrário o gráfico da aceleração nesse eixo terá parecenças com o gráfico da aceleração da figura 3.2b.

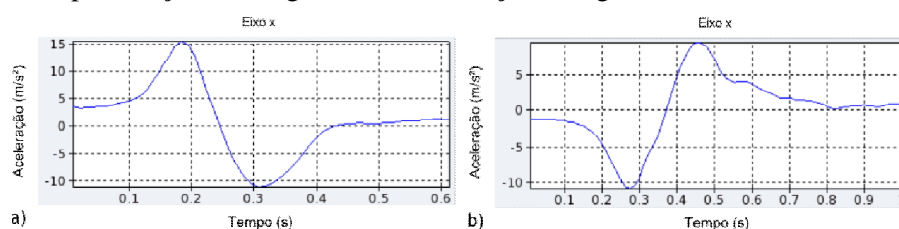


Figura 3.2 - Exemplo típico das acelerações lidas pelo acelerómetro do eixo x quando o movimento feito é: a) linear para a esquerda, b) linear para a direita

Para identificar o eixo onde um movimento linear se dá, basta comparar as diferenças de amplitudes nos pontos máximos e mínimos em relação à média nos três eixos, sendo o eixo do movimento aquele onde essa diferença é maior, como se pode ver nas figuras 3.3a, 3.3b e 3.3c onde esta diferença de amplitude é maior no eixo x, logo o movimento foi feito segundo esse eixo.

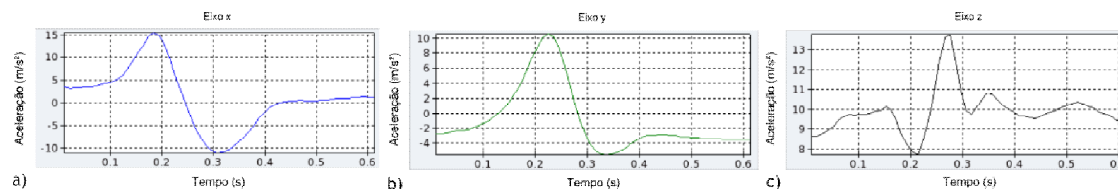


Figura 3.3 - Exemplo das acelerações lidas nos três eixos do acelerómetro num movimento linear para a esquerda: a) eixo x, b) eixo y, c) eixo z

Comparando os movimentos lineares sobre um eixo com os movimentos circulares no plano xz expressos nas figuras 3.4a, 3.4b e 3.4c, chega-se à conclusão que é possível diferenciá-los através da análise do número de máximos e de mínimos da aceleração em cada eixo, uma vez que um movimento circular pode ser visto como uma combinação de movimentos sucessivos em cada eixo e simultâneos nos eixos x e z. O sentido destes movimentos é detectável através da análise da sequência de movimentos nos eixos x e z.

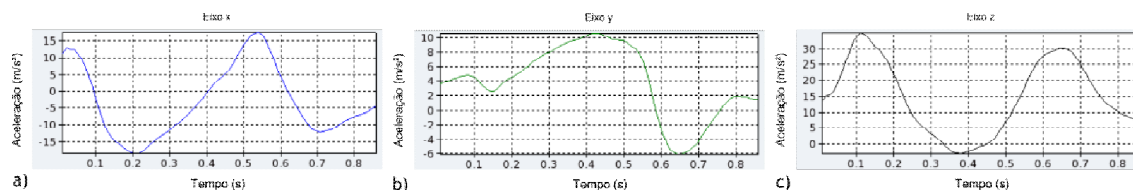


Figura 3.4 - Exemplo das acelerações lidas quando o é feito um movimento circular no plano xz no sentido horário: a) eixo x, b) eixo y, c) eixo z

No Anexo 1 estão apresentados os gráficos das acelerações correspondentes a todos os movimentos analisados.

### 3.2 Receptor de Infra-Vermelhos

O *Wiimote* está também dotado de um sensor infra-vermelho capaz de efectuar *tracking* de quatro fontes luminosas em simultâneo. Para ser possível calcular a distância e ângulo a que o ponto médio de duas fontes luminosas se encontra do *Wiimote* foram determinados experimentalmente os seus ângulos de abertura vertical ( $\gamma$ ) e horizontal ( $\phi$ ). Para determinar estas aberturas colocaram-se duas fontes luminosas a uma distância conhecida ( $d$ ) e aproximou-se o *Wiimote* na direcção perpendicular à recta formada pelos dois pontos luminosos, sendo a intercepção destas duas rectas o ponto médio desta última recta. Quando o *Wiimote* deixar de detectar qualquer ponto luminoso é medida a distância entre este e o ponto médio dos focos de luz ( $h$ ). O ângulo de abertura é dado pela equação 3.1, bastando rodar o *Wiimote* sobre o seu eixo  $y$  e repetir o procedimento para medir a outra abertura. Os resultados obtidos para  $\phi$  e  $\gamma$  foram  $43.8^\circ$  e  $33.5^\circ$ , respectivamente. A resolução deste sensor é  $1024 \times 768$ .

$$\phi = \arctan \frac{d}{h} \quad (3.1)$$

No decorrer da utilização deste sensor foi descoberta uma faixa onde novas fontes luminosas não eram detectadas mas fontes luminosas detectadas previamente não eram perdidas, sendo-lhe dado o nome de faixa de não detecção. Assumindo que a figura 3.5 ilustra a imagem captada pelo sensor infra-vermelho num determinado instante, o par luminoso 1 encontra-se totalmente no interior da imagem, logo ambas as fontes luminosas são detectadas pelo *Wiimote*. Se este par for deslocado para a posição 2 ambas as fontes continuarão a ser detectadas, apesar de uma estar no interior da imagem (zona branca) e outra no interior da faixa mencionada anteriormente (zona cinza). Se aparecer um novo par luminoso (par 3) apenas o ponto luminoso à direita será detectado sendo necessário deslocar este par para o interior da imagem para o *Wiimote* detectar o ponto da esquerda. Esta faixa tem uma largura de cerca de 10 *pixéis* e altura de 5 *pixéis*.

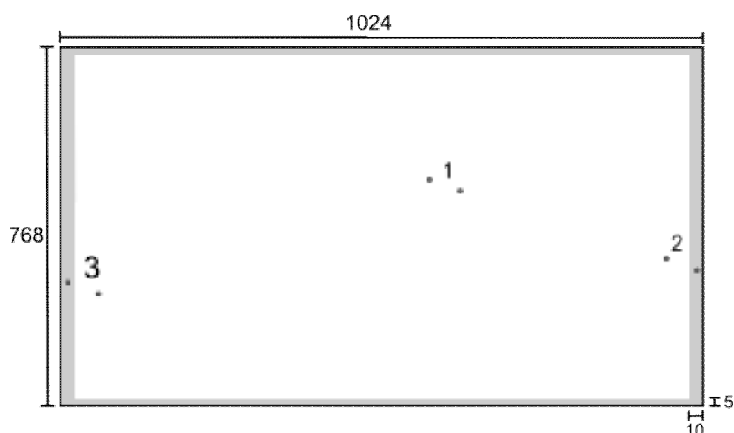


Figura 3.5 - Representação da faixa de não detecção do sensor infra-vermelho

### 3.3 Restantes Constituintes

A comunicação entre o *Wiimote* e a consola *Wii* é feita através de *Bluetooth*, sendo possível conectar até quatro controladores em simultâneo. Esta característica confere ao *Wiimote* uma enorme liberdade de movimentos já que este tipo de comunicação é *wireless* e o seu raio de operação é de várias dezenas de metros, o que combina perfeitamente com a sensorização de movimentos que o tornou famoso.

Por fim resta mencionar o *Wiimote* contém onze botões de pressão, quatro *LEDs* cuja função é indicar o estado da bateria o que mostra que podem ser um elemento importante na interface com o utilizador e um *speaker*.

### 3.4 Conclusões

Neste capítulo foram apresentados os constituintes do *Wiimote*, sendo seu acelerómetro tridimensional e o seu receptor de infra-vermelhos aqueles que mais se destacam. A exploração das características do primeiro possibilitou o levantamento do seu sistema de eixos interno e a detecção das características das acelerações que permitiam diferenciar diversos movimentos, tendo particular interesse aqueles que permitiam diferenciar a direcção e sentido de movimentos lineares e o sentido de movimentos circulares. Para o receptor de infra-vermelhos foi levantada a sua resolução e os seus ângulos de abertura vertical e horizontal. Este último será especialmente importante na execução do modo de controlo baseado neste sensor.

# Capítulo 4

## Solução Proposta

### 4.1 Introdução

Este trabalho foi desenvolvido num ambiente Linux de forma a facilitar o transporte para o robô de limpeza e baseia-se no funcionamento em simultâneo de duas aplicações com objectivos diferentes. A primeira tinha como objectivo estabelecer uma ligação entre o *Wiimote* e o *PC* permitindo à segunda focar-se sobre os algoritmos e as diferentes formas de controlo. Estas aplicações chamam-se “Aplicação de Interface” e “Aplicação de Controlo”, respectivamente.

### 4.2 Estrutura da Solução Proposta

Na figura 4.1 está apresentada a estrutura proposta para a resolução do problema apresentado neste trabalho, criar uma interface homem-máquina, ou, mais especificamente, interface homem-robô, baseada na utilização do *Wiimote*.

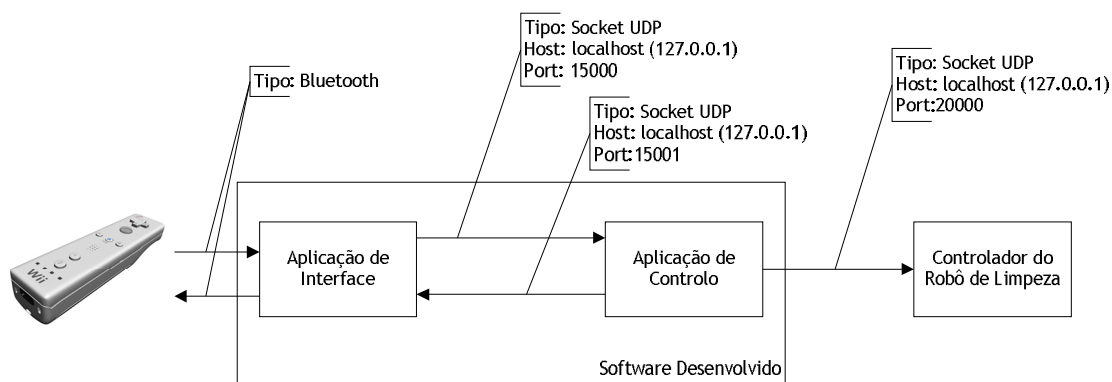


Figura 4.1 - Diagrama de funcionamento geral do software desenvolvido

Todas as comunicações são feitas através de *sockets* UDP excepto a comunicação entre o *Wiimote* e o *PC* que necessita de ser feita através de *Bluetooth*. A escolha deste tipo de

comunicações deve-se à sua facilidade de utilização graças à existência de diversas bibliotecas standard. A escolha do protocolo UDP em detrimento do protocolo TCP deve-se ao facto do seu foco ser a mensagem, enquanto o foco dos *sockets* TCP é a ligação entre aplicações, o que poderia por em causa os requisitos de controlo em tempo real do sistema.

### 4.3 Conclusões

Neste capítulo foi apresentada a solução proposta para o problema deste trabalho, passando essa solução pela implementação de duas aplicações que comunicam via *sockets* UDP. A comunicação com o controlador é feita através do mesmo meio de comunicação. No capítulo seguinte serão apresentados os pormenores do desenvolvimento da Aplicação de Interface e da Aplicação de Controlo e dos modos de funcionamento implementados nesta última.



# Capítulo 5

## Software Desenvolvido

### 5.1 Introdução

Neste capítulo serão apresentados os detalhes da implementação das duas aplicações mencionadas na estrutura da solução proposto, apresentada no capítulo anterior. Estas duas aplicações têm como nome Aplicação de Interface e Aplicação de Controlo, respectivamente. A divisão da interface de controlo desenvolvida nestas duas aplicações representa uma repartição de funções. A primeira aplicação trata da comunicação bilateral entre o PC e o Wiimote e da interpretação dos relatórios enviados por este, permitindo à segunda focar-se sobre os modos de funcionamento presentes na interface, os seus métodos de controlo e a actuação sobre o controlador do robô.

### 5.2 Aplicação de Interface

Esta aplicação foi desenvolvida no ambiente KDevelop C/C++ [22] em linguagem C com o objectivo de tirar partido da biblioteca *libcwiimote* [23]. O recurso a esta biblioteca trás a vantagem da interpretação dos relatórios do comando guardados na *stack* de *Bluetooth* ser feita de forma automática e dos dados de interesse serem guardados em variáveis com nomes apropriados. Para utilizar esta biblioteca o sistema operativo utilizado necessita de ser Linux e de ter instalado as bibliotecas relativas às comunicações *Bluetooth*, *Bluez* [24].

#### 5.2.1 Utilização da biblioteca libcwiimote

De seguida são apresentadas as funções fornecidas por esta biblioteca que se revelaram as mais importantes para o desenvolvimento desta aplicação:

- A função **wiimote\_connect()** permite conectar o *Wiimote* ao PC a partir do seu endereço, ficando este associado a uma estrutura do tipo *wiimote\_t* responsável por guardar todos os dados associados aos diferentes constituintes do *Wiimote*;

- A função **wiimote\_disconnect()** efectua a operação contrária à função anterior, isto é desliga a ligação entre o *Wiimote* e o *PC* e limpa todas as variáveis que a ele estavam associadas;
- **wiimote\_is\_open()** é uma função booleana que permite verificar o estado da ligação entre o *Wiimote* e o *PC*, o que a torna adequada para ser utilizada num ciclo infinito, tal como um **while()**;
- Se a ligação estiver estabelecida a função **wiimote\_update()** permite actualizar as diferentes variáveis associadas ao *Wiimote* lido, obtendo-se assim os valores da amostra mais recente.

A estas funções estão associadas três grandes estruturas aonde estão incorporadas as variáveis relativas aos botões, aos valores dos acelerómetros e dos pontos infra-vermelhos detectados. Estas estruturas são:

- **wiimote.axis** é a estrutura associada aos acelerómetros e é constituída por três variáveis de 8bit cada. Estas variáveis representam as acelerações em cada um dos eixos;
- A estrutura **wiimote.irk**,  $k=1..4$ , é composta por três variáveis associadas ao ponto infra-vermelho definido por  $k$ . As variáveis  $x$  e  $y$  têm tamanho 10bit e a variável  $size$  4bit;
- A estrutura **wiimote.keys** engloba as variáveis associadas ao estado dos botões.

Por fim são necessários activar dois modos de funcionamento do *Wiimote* de forma que este inclua certos dados nos relatórios que envia. Estes modos são:

- **wiimote.mode.acc** – este modo activa o envio da informação referente aos acelerómetros que é posteriormente guardada na estrutura **wiimote.axis**;
- **wiimote.mode.ir** – este modo activa o envio de informação associada aos pontos infra-vermelhos que o *Wiimote* detecta, sendo estes dados guardados nas quatro estruturas **wiimote.ir**;

### 5.2.2 Estabelecimento da comunicação entre aplicações

Tendo conhecimento das funções, variáveis e modos necessários tornou-se possível elaborar uma aplicação capaz de retirar todas as informações necessárias do *Wiimote* e fornecê-las à aplicação de controlo. Para tal foi necessário criar um protocolo de comunicações, sendo escolhido o envio de uma trama através de *sockets* UDP. A comunicação é feita nos dois sentidos, isto é, da aplicação de interface para a aplicação de controlo e desta para a aplicação de interface. Do ponto de vista da aplicação de interface a trama a enviar é composta por dois *start byte*, pelo vector *Data* e por um *stop byte* e a trama a receber é composta por dois *start byte*, por um vector de comando (*Cmd*) e por um *stop byte*, como está representado nas figuras 5.1 e 5.2, respectivamente.

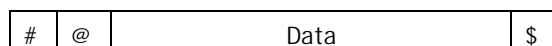


Figura 5.1 - Estrutura da trama a enviar pela Aplicação de Interface

O vector *Data* é composto pelo valor das variáveis das estruturas apresentadas necessárias à aplicação de controlo. Na lista seguinte são apresentadas as variáveis utilizadas nesta trama, sendo o número de apresentação na lista equivalente ao número do *byte* do vector *Data* onde essa variável se encontra:

1. Valor dos primeiros 8bit da estrutura **wiimote.keys**;
2. Valor dos últimos 8bit da estrutura **wiimote.keys**;
3. Aceleração ao longo do eixo x do *Wiimote* (**wiimote.axis.x**);
4. Aceleração no eixo y do *Wiimote* (**wiimote.axis.y**);
5. Aceleração no eixo z do *Wiimote* (**wiimote.axis.z**);
6. Byte mais significativo de **wiimote.ir1.x**;
7. Byte menos significativo de **wiimote.ir1.x**;
8. Byte mais significativo de **wiimote.ir1.y**;
9. Byte menos significativo de **wiimote.ir1.y**;
10. **wiimote.ir1.size**;
- 11 a 15. Equivalente a 6 a 10 mas para a estrutura **wiimote.ir2**;
- 16 a 20. Equivalente a 6 a 10 mas para a estrutura **wiimote.ir3**;
- 21 a 25. Equivalente a 6 a 10 mas para a estrutura **wiimote.ir4**;

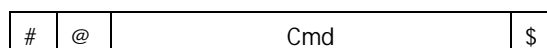


Figura 5.2 - Estrutura da trama a receber pela Aplicação de Interface

A existência do vector *Cmd* deve-se à impossibilidade da aplicação de controlo comunicar directamente com o *Wiimote*, sendo necessário que esta envie este vector de comandos para a aplicação de interface para que esta possa provocar no *Wiimote* as acções desejadas. Este vector é composto por dois *bytes* cuja função é:

1. Indicar o valor desejado para a estrutura **wiimote.led** responsável pelo estado dos quatro *LEDs* e da vibração do *Wiimote*, não sendo esta utilizada por motivos de conservação das baterias. A utilização dos *LEDs* tem como objectivo fornecer uma indicação visual ao utilizador do modo de operação da aplicação de controlo que se encontra activo;
2. O segundo *byte* tem como objectivo estabelecer comandos de baixo nível a serem executados dentro da aplicação de interface, como por exemplo terminar esta aplicação quando a aplicação de controlo também termina.

### 5.2.3 Descrição de Funcionamento

Tendo todos os aspectos do protocolo de comunicação definidos, foi possível criar a aplicação de interface desejada. Esta aplicação começa por estabelecer ligação com o *Wiimote* através da função **wiimote.connect()** e se for bem sucedida cria os *sockets* servidor e cliente, estando esta aplicação associada ao primeiro e a aplicação de controlo associada ao segundo. Caso as operações anteriores sejam bem sucedidas os modos acelerómetro e infra-vermelhos são activados através dos comandos **wiimote.mode.acc=1** e **wiimote.mode.ir=1**, respectivamente, entrando o programa num ciclo infinito controlado pela função **wiimote\_is\_open()** e dentro do qual acontece o refrescamento dos valores das variáveis através da função **wiimote.update()**, a criação do vector *Data*, o envio da trama para a aplicação de controlo, o recebimento da trama de resposta e a actuação sobre os *LEDs* do *Wiimote* conforme os dados contidos no vector *Cmd*. Este ciclo infinito termina quando a ligação *Wiimote-PC* é

terminada através da função **wiimote.disconnect()**, podendo isso se dever ao encerramento da aplicação de controlo, ao *timeout* da ligação entre as duas aplicações ou ao premir do botão *Home* (**wiimote.keys.home = 1**).

O funcionamento desta aplicação está demonstrado no fluxograma da figura 5.3.

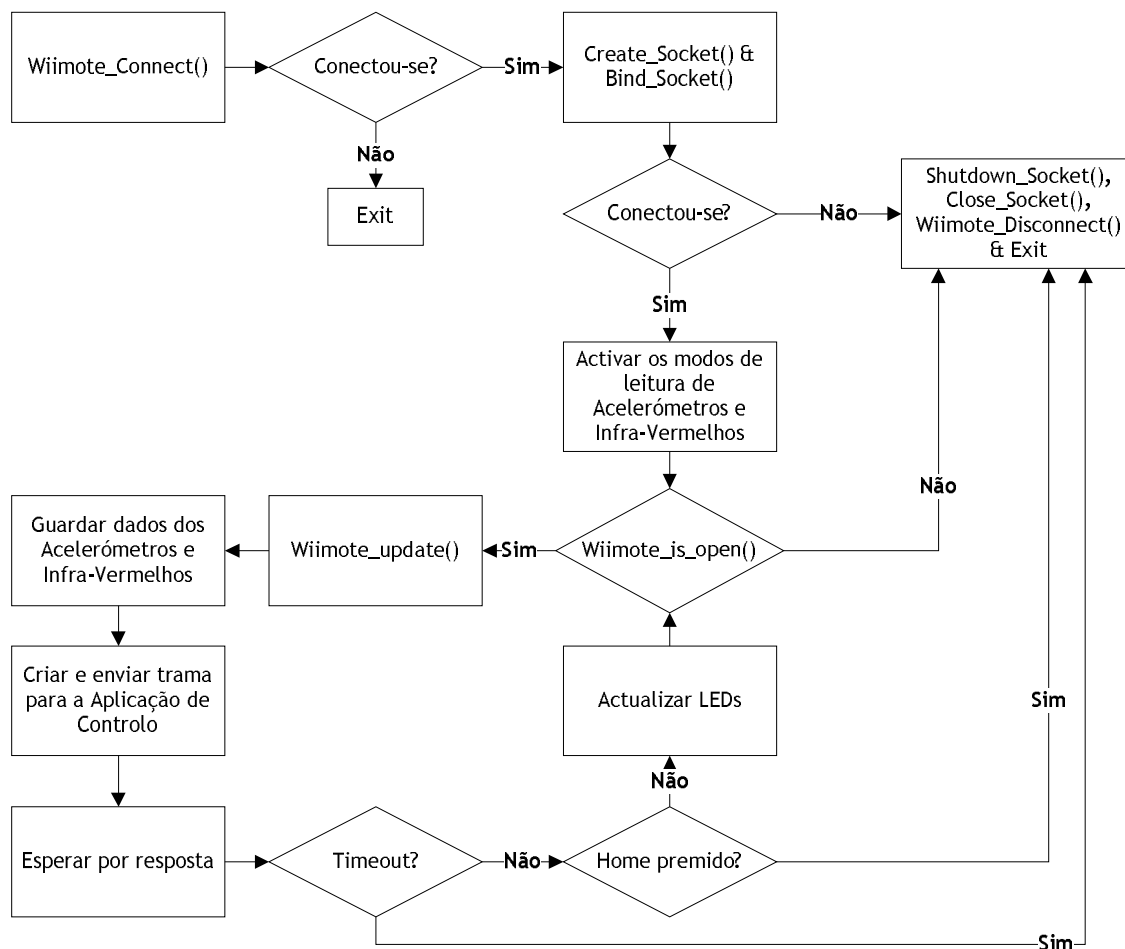


Figura 5.3 - Diagrama de fluxo da Aplicação de Interface

## 5.3 Aplicação de Controlo

Na secção anterior foram apresentados os pormenores da Aplicação de Interface cujo papel no sistema implementado é o de intermediário entre o *Wiimote* e a Aplicação de Controlo. Esta aplicação é o centro do sistema desenvolvido, estando nela concentrados os diversos modos de funcionamento que permitirão ao utilizador interagir com o robô.

O ambiente de desenvolvimento escolhido para esta aplicação foi *Lazarus Free Pascal* [25] devido ao seu carácter gratuito e, sobretudo, a ser o ambiente de programação onde o programa de controlo do robô foi desenvolvido. Todos os componentes utilizados nesta aplicação foram *standard* com excepção do componente que permite estabelecer comunicações através de *sockets*, *INet*. Este componente permite que todo o processo de controlo seja desencadeado pela recepção de uma nova trama vinda da aplicação de interface estabelecendo sincronismo entre as duas aplicações.

Esta aplicação tem como objectivo implementar diversos modos de utilização do *Wiimote* tendo como objectivo aplicá-los no sistema robótico disponível. Estes modos de utilização resultam de uma combinação dos três grandes elementos presentes no *Wiimote*, os botões, o acelerómetro tridimensional e o receptor de infra-vermelhos, e para todos eles foi analisada a sua compatibilidade com diversas técnicas de controlo conhecidas. De seguida são apresentados os modos implementados, as razões que levaram à escolha da sua técnica de controlo e o seu funcionamento de uma forma detalhada.

### 5.3.1 Funcionamento como Volante

#### 5.3.1.1 Princípio de Funcionamento

Este modo de funcionamento tem como objectivo simular o volante e os pedais de aceleração e travagem de um automóvel recorrendo ao *Wiimote*. O princípio de funcionamento deste modo é detectar a inclinação do *Wiimote* e através dela actuar sobre as velocidades linear e angular do robô, podendo a inclinação ser horizontal, vertical ou ambas. A inclinação pode ser medida considerando que no momento da amostragem o *Wiimote* se encontra parado sendo a única actuação sofrida pelos acelerómetros devida à aceleração da gravidade.

O primeiro ponto necessário à elaboração deste modo de funcionamento é definir a posição de actuação nula, isto é, a posição do *Wiimote* que anula as velocidades linear e angular. Por comparação com um volante tradicional esta posição é a demonstrada pela figura 5.4. Com o *Wiimote* nesta posição e atendendo ao seu sistema de eixos (figura 3.1) a aceleração da gravidade é lida pelo acelerómetro do eixo x. Ao inclinar o *Wiimote* horizontalmente, isto é, para a esquerda ou direita, a medição da aceleração da gravidade passa a ser repartida pelos acelerómetros dos eixos x e y (figura 5.5) ao incliná-lo verticalmente, para a frente ou para trás, esta passa a ser lida pelos acelerómetros dos eixos x e z (figura 5.6), ou seja, lendo os valores da aceleração segundo os eixos y e z é possível identificar qual a inclinação do comando.

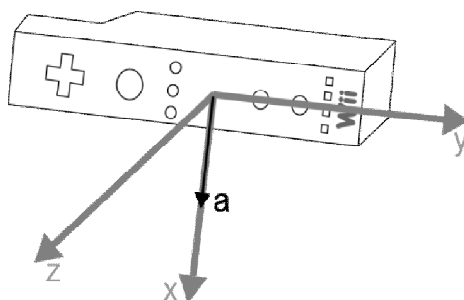


Figura 5.4 - Posição de actuação nula

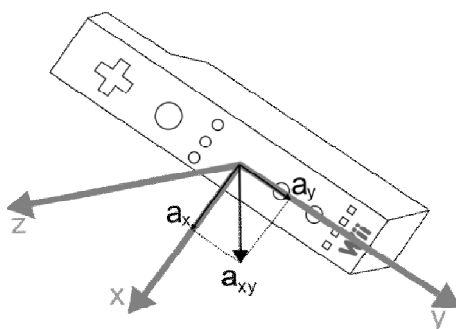


Figura 5.5 - Repartição da gravidade quando o Wiimote está inclinado horizontalmente

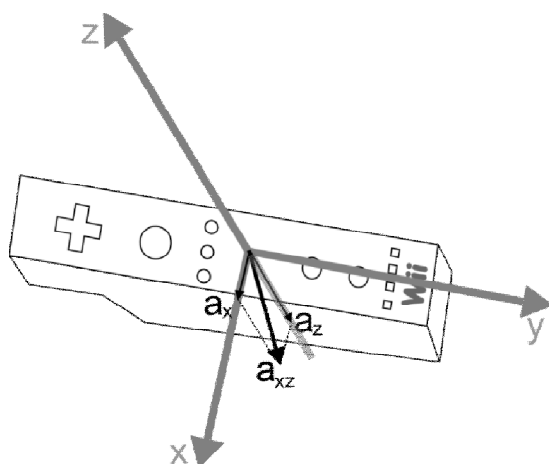


Figura 5.6 - Repartição da gravidade quando o Wiimote está inclinado verticalmente

Com base nesta formulação teórica definiram-se nove pontos de controlo possíveis, sendo para cada um deles medido as acelerações em cada eixo, estando estas expressas na tabela 5.1 pela ordem aceleração segundo x, segundo y e segundo z. Os resultados obtidos foram os seguintes:

Tabela 5.1 - Valores das acelerações lidas nos nove pontos de controlo definidos

|                       |                      |                      |
|-----------------------|----------------------|----------------------|
| Frente e Esquerda     | Frente e CentroH     | Frente e Direita     |
| 119                   | 112                  | 116                  |
| 150                   | 133                  | 118                  |
| 146                   | 150                  | 150                  |
| Esquerda e CentroV    | CentroV e CentroH    | Direita e CentroV    |
| 114                   | 104                  | 109                  |
| 152                   | 132                  | 115                  |
| 135                   | 131                  | 136                  |
| Retaguarda e Esquerda | Retaguarda e CentroH | Retaguarda e Direita |
| 117                   | 110                  | 112                  |
| 150                   | 131                  | 123                  |
| 117                   | 115                  | 116                  |

A notação CentroV (centro vertical) indica o facto do *Wiimote* não estar inclinado para a frente ou para trás, enquanto a notação CentroH (centro horizontal) indica que o *Wiimote* não está inclinado para a esquerda ou para a direita. Repare-se que a aceleração medida segundo o eixo x se mantém praticamente constante ao longo de toda a tabela, enquanto a aceleração segundo o eixo y varia de coluna para coluna e a aceleração segundo z varia de linha para linha, validando a formulação teórica anteriormente exposta, isto é, considerando apenas as acelerações medidas nos eixos y e z é possível distinguir os nove pontos de controlo considerados.

### 5.3.1.2 Aplicação do Algoritmo de Controlo Difuso

O método de controlo escolhido para este modo de funcionamento foi o controlo por lógica difusa, devido a ser um método fácil de implementar e de actuação rápida e, sobretudo, devido à dificuldade de determinar os limites entre os pontos de controlo considerados e ao carácter não-

linear subjacente à análise das acelerações, o que inviabiliza as técnicas de controlo clássico. Outras técnicas de controlo inteligente poderiam ser utilizadas mas devido ao baixo número de entradas e situações a analisar (regras) o controlo baseado em lógica difusa é a melhor escolha. A topologia escolhida foi Mamdani com duas entradas, acelerações medidas nos eixos y e z, e duas saídas, valores de actuação sobre as velocidades angular e linear. As suas características são: Cada variável é composta por três conjuntos difusos triangulares simétricos em relação ao valor central; A base de regras criada é composta pelas seis regras abaixo apresentadas (tabela 5.2), cada uma referente a uma linha ou coluna da tabela 5.1; O cálculo do grau de satisfação de cada regra é feito utilizando o operador mínimo para as expressões ‘E’ e o operador máximo para as expressões ‘OU’; A agregação das saídas das regras é feita através do operador máximo; A *desfuzificação* é feita através do centro de gravidade.

Tabela 5.2 - Base de regras utilizada

|  |
|--|
| SE Frente E (Esquerda OU Direita OU CentroH) ENTÃO $v_{\text{Positivo}}$ ;         |
| SE CentroV E (Esquerda OU Direita OU CentroH) ENTÃO $v_{\text{Nulo}}$ ;            |
| SE Retaguarda E (Esquerda OU Direita OU CentroH) ENTÃO $v_{\text{Negativo}}$ ;     |
| SE Esquerda E (Frente OU Retaguarda OU CentroV) ENTÃO $\omega_{\text{Positivo}}$ ; |
| SE CentroH E (Frente OU Retaguarda OU CentroV) ENTÃO $\omega_{\text{Nulo}}$ ;      |
| SE Direita E (Frente OU Retaguarda OU CentroV) ENTÃO $\omega_{\text{Negativo}}$ ;  |

### 5.3.1.3 Descrição de Funcionamento

A interface referente a este modo de funcionamento está localizada na página *Fuzzy Steering* da aplicação de controlo. O início deste modo de funcionamento dá-se após ser recebida uma nova trama da aplicação de interface e o botão ‘A’ ser premido. Se tal acontecer é verificada a estabilidade dos valores das acelerações lidas nos eixos x, y e z, para garantir que o controlo difuso apenas acontece quando apenas a aceleração da gravidade está a ser lida. Esse teste de estabilidade é feito através da comparação do valor actual de cada uma das acelerações com a média das cinco amostras anteriores. Se as três acelerações lidas forem consideradas estáveis dá-se início ao ciclo de controlo *Fuzzy*, caso contrário o programa de controlo espera por uma nova trama da aplicação de interface. Em qualquer um dos casos é actualizado o vector das últimas cinco amostras da aceleração em cada um dos eixos.

A primeira acção realizada após o início do ciclo de controlo *Fuzzy* é calcular os valores de pertença das duas variáveis de entrada em cada um dos seus conjuntos difusos. Isto é feito através da desagregação de cada um dos triângulos que definem um conjunto difuso em duas rectas, sendo o valor de pertença da variável em causa o menor valor das coordenadas das duas rectas se o valor da abcissa utilizado for o valor da variável, como está representado na figura 5.7.

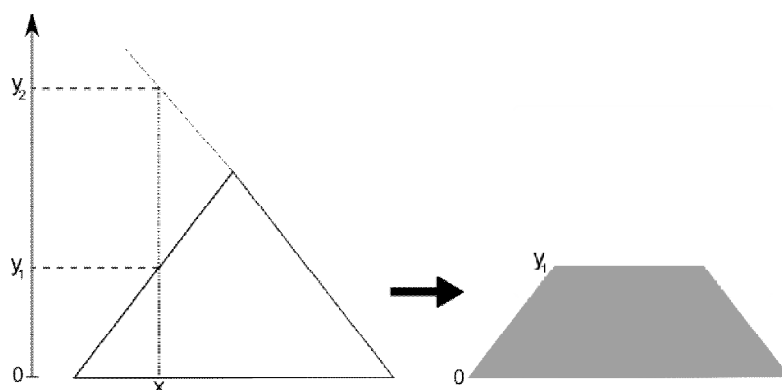


Figura 5.7 - Cálculo do valor de pertença de uma variável de valor x num conjunto difuso

De seguida são calculados os valores de disparo de cada regra através dos operadores mínimo, para as expressões *AND* (E), e máximo, para as expressões *OR* (OU). Considerando que as variáveis utilizadas são y e z e os seus valores de pertença nos conjuntos positivo, nulo e negativo são  $P_{y,P}$ ,  $P_{y,Z}$  e  $P_{y,N}$  e  $P_{z,P}$ ,  $P_{z,Z}$  e  $P_{z,N}$ , respectivamente, a saída de cada regra da tabela 5.2 é calculada através das expressões da tabela 5.3.

Tabela 5.3 - Expressões de cálculo do nível de disparo das regras

|   |
|---|
| $\text{Min}( P_{z,P} , \text{Max}( P_{y,P} , P_{y,Z} , P_{y,N} ) ) ;$ |
| $\text{Min}( P_{z,Z} , \text{Max}( P_{y,P} , P_{y,Z} , P_{y,N} ) ) ;$ |
| $\text{Min}( P_{z,N} , \text{Max}( P_{y,P} , P_{y,Z} , P_{y,N} ) ) ;$ |
| $\text{Min}( P_{y,P} , \text{Max}( P_{z,P} , P_{z,Z} , P_{z,N} ) ) ;$ |
| $\text{Min}( P_{y,N} , \text{Max}( P_{z,P} , P_{z,Z} , P_{z,N} ) ) ;$ |
| $\text{Min}( P_{y,N} , \text{Max}( P_{z,P} , P_{z,Z} , P_{z,N} ) ) ;$ |

O passo seguinte é efectuar a agregação das saídas das regras de forma a ser possível calcular os valores das saídas do sistema difuso. Para tal foram criados previamente três conjuntos difusos triangulares para cada variável de saída cujos valores centrais dos conjuntos positivo, nulo e negativo são 1, 0 e -1, respectivamente. Estes valores justificam-se por se pretender manter as velocidades  $v$  e  $\omega$  nos intervalos  $[v_{nom}, -v_{nom}]$  e  $[\omega_{nom}, -\omega_{nom}]$ , respectivamente, sendo da responsabilidade da aplicação de controlo do robô definir os valores de  $v_{nom}$  e  $\omega_{nom}$ . Para simplificar a agregação das saídas das regras definiu-se que cada conjunto difuso das saídas seria composto por cem *singletons* cujo valor era definido pelas rectas envolventes do triângulo. A simplificação vem do facto de se poder calcular a saída em cada *singleton* como sendo o conjunto de duas operações máximo e mínimo. A primeira operação de agregação é um mínimo entre o valor de disparo da regra onde o conjunto difuso a que pertence esse *singleton* é saída e o valor máximo desse *singleton*. A segunda operação é um máximo entre *singletons* de conjuntos diferentes mas localizados na mesma abcissa. Estas operações de simplificação estão exemplificadas na figura 5.8.



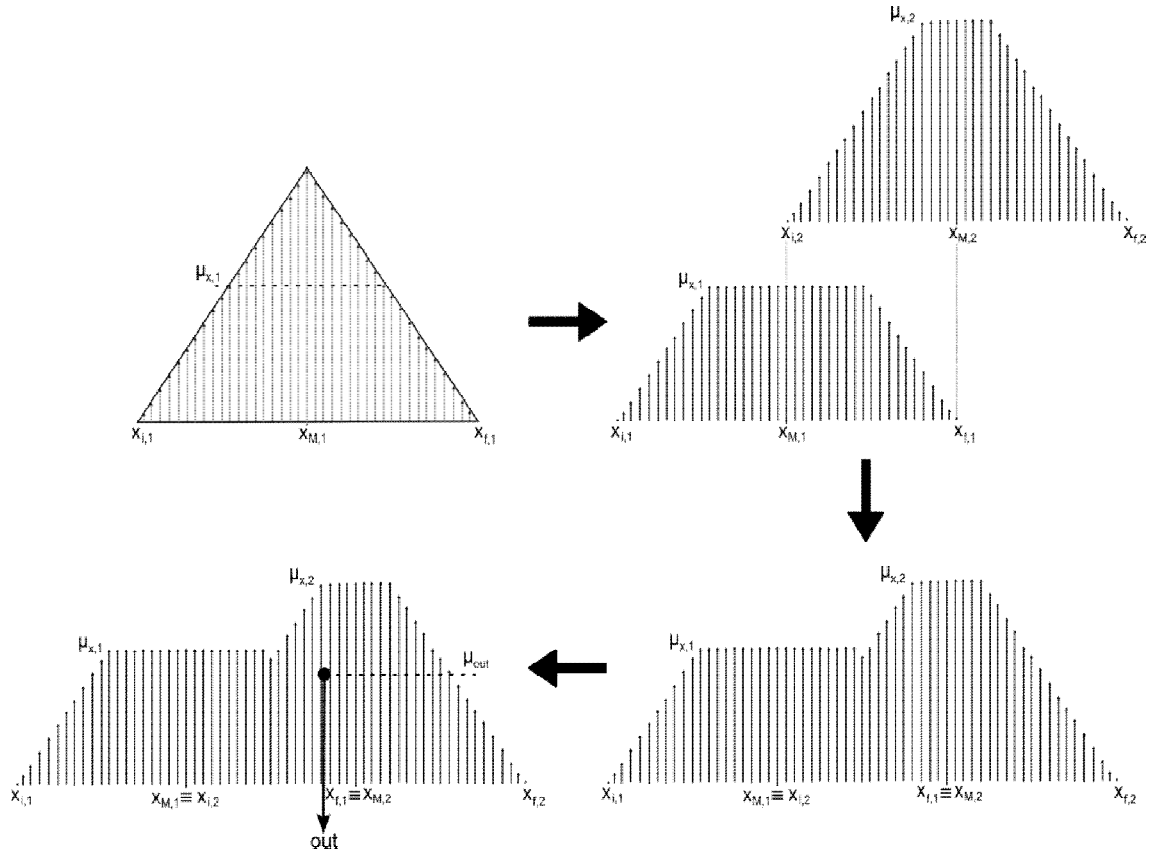


Figura 5.8 - Representação gráfica das operações realizadas sobre os singletons que definem os conjuntos difusos das saídas

Por fim dá-se a *desfuzificação* para obter as duas saídas. O método utilizado foi o centro de gravidade já que é bastante simples de utilizar em conjunto com os *singletons*. Desta forma a saída é calculada através de uma média pesada dos valores das abcissas de cada *singleton* tendo em conta o seu valor de pertinência (equação 5.1), onde  $\mu_k$  é o valor de pertinência do *singleton* localizado em  $x_k$  e *out* representa  $v_{out}$  e  $w_{out}$ , dependendo da saída que estiver a ser calculada.

$$out = \frac{\sum_{k=1}^{200} \mu_k \cdot x_k}{\sum_{k=1}^{200} \mu_k} \quad (5.1)$$

Terminado o ciclo *Fuzzy* os valores de actuação sobre  $v_{nom}$  e  $\omega_{nom}$  são enviados para o programa de controlo do robô através de um protocolo de comunicação entre as duas aplicações baseado em *sockets* UDP, tal como a comunicação entre a aplicação de interface e a aplicação de controlo. Este ciclo de controlo repete-se até o botão ‘A’ ser premido novamente ou até ser escolhido outro modo de funcionamento pelo utilizador.

Neste modo de funcionamento existe ainda a possibilidade de configurar os valores dos conjuntos difusos das entradas. Para tal é necessário premir o botão ‘1’ para entrar no modo de treino e escolher as posições desejadas para os conjuntos centro, frente, retaguarda, esquerda e direita, respectivamente, validando a escolha através do botão ‘1’, sendo possível cancelar o modo de treino através do botão ‘2’.

Na figura 5.9 está representado o fluxograma deste modo de funcionamento.

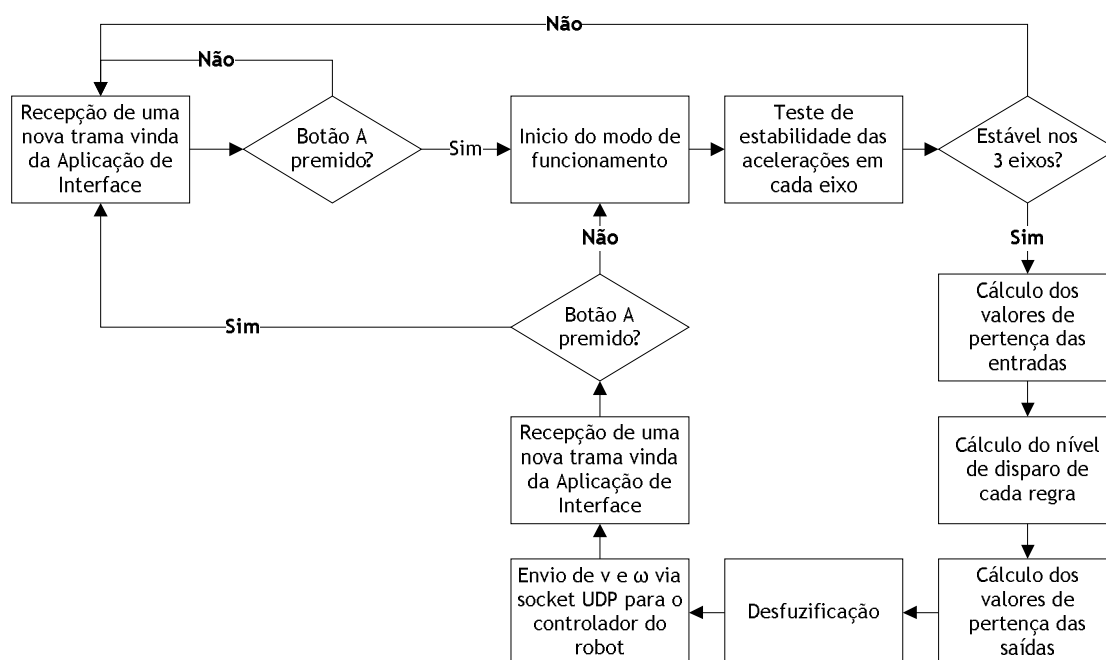


Figura 5.9 - Diagrama de fluxo do funcionamento como volante

## 5.3.2 Monitorização de Infra-Vermelhos

### 5.3.2.1 Princípio de Funcionamento

Este modo de funcionamento tem como objectivo tirar partido das capacidades de captação de fontes infra-vermelhas por parte do *Wiimote* para determinar distâncias e ângulos que possam ser utilizadas de uma forma activa, através da actuação sobre as velocidades linear e angular, ou passiva, através da incorporação destes dados num algoritmo de auto-localização. Estas duas formas de utilização foram criadas considerando que o *Wiimote* se encontra acoplado à estrutura do robô e a estrutura de fontes luminosas (2 *LEDs*) se encontra colocada numa parede ou superfície externa ou na mão do utilizador.

### 5.3.2.2 Descrição de Funcionamento

Os requisitos necessários para calcular as distâncias  $r_1$  e  $r_2$  e por consequência a distância  $r$  e o ângulo  $\theta$  representados na figura 5.10 são o conhecimento prévio da distância entre os dois *LEDs* ( $d$ ) e do ângulo de abertura horizontal ( $\varphi=43.8^\circ$ ). Para calcular a distância  $r_1$  o primeiro passo é calcular o ponto de intercepção entre a recta gerada pelos dois pontos luminosos detectados e a recta vertical  $x=512$ , tendo este ponto as coordenadas  $(x_c, y_c)$ . O valor  $x=512$  explica-se devido a ser metade da resolução horizontal do sensor infra-vermelho, o que faz que esta recta divida a imagem recebida por este sensor em duas metades, facilitando cálculos futuros.

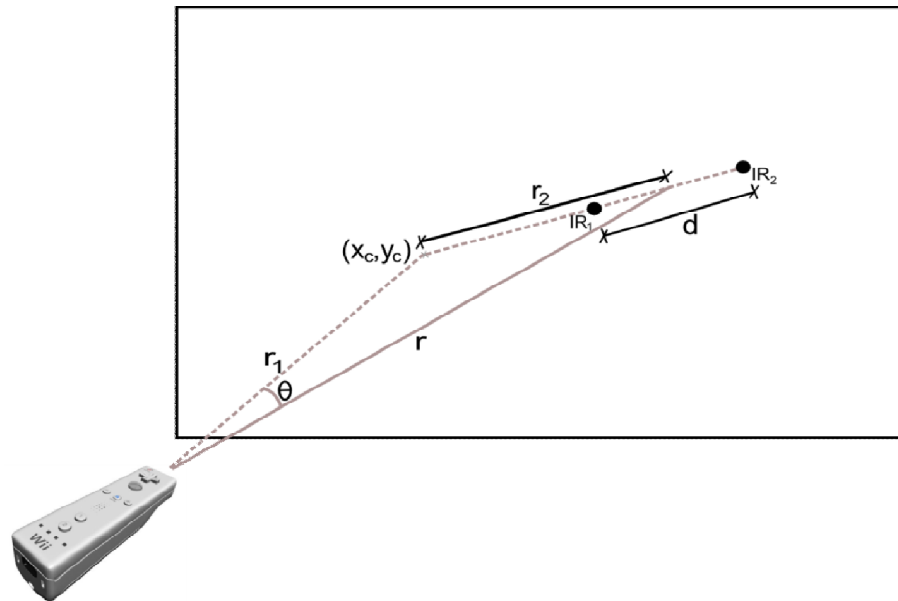


Figura 5.10 - Esquema das distâncias utilizadas neste modo de funcionamento.

De seguida calculam-se as distâncias  $p_1$  e  $p_2$ , em *pixels*, entre os pontos luminosos e o ponto  $(x_c, y_c)$  e a distância  $p$  entre os dois pontos luminosos através do teorema de Pitágoras (equações 5.2, 5.3 e 5.4), estando estas distâncias representadas na figura 5.11.

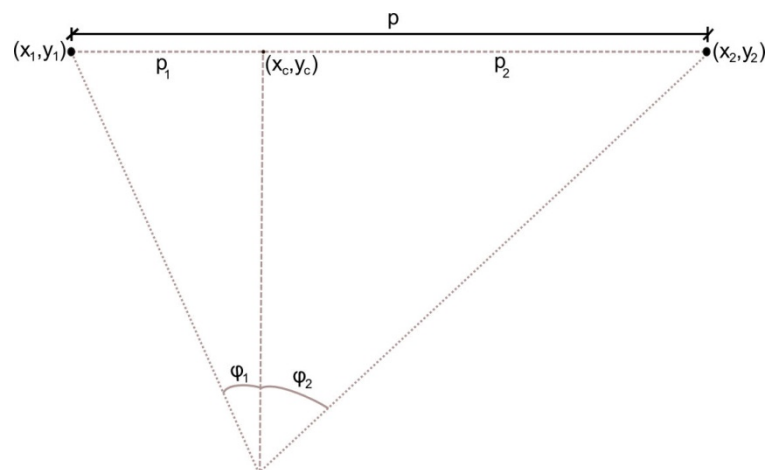


Figura 5.11 - Esquema planar das distâncias dos pontos detectados ao centro (em pixel) e dos ângulos desses pontos em relação ao ponto do Wiimote.

$$p_1 = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2} \quad (5.2)$$

$$p_2 = \sqrt{(x_2 - x_c)^2 + (y_2 - y_c)^2} \quad (5.3)$$

$$p = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5.4)$$

O passo seguinte é calcular os ângulos  $\varphi_1$  e  $\varphi_2$ . Estes podem ser obtidos através da relação matemática entre a distância máxima de um ponto luminoso ao ponto  $(x_c, y_c)$ , 512 *pixéis*, a abertura horizontal que um ponto tem quando está colocado a essa distância,  $\varphi/2$ , e as distâncias  $p_1$  e  $p_2$  obtidas no passo anterior. Esta relação está expressa nas equações 5.5 e 5.6.

$$\varphi_1 = \frac{p_1 \cdot \varphi}{1024} \quad (5.5)$$

$$\varphi_2 = \frac{p_2 \cdot \varphi}{1024} \quad (5.6)$$

Por analogia com o passo anterior as distâncias  $d_1$  e  $d_2$ , equivalentes a  $p_1$  e  $p_2$  mas em centímetros, podem ser obtidas por comparação entre as distâncias  $d$  e  $p$ , sendo esta relação expressa pelas equações 5.7 e 5.8.

$$d_1 = \frac{p_1 \cdot d}{p} \quad (5.7)$$

$$d_2 = \frac{p_2 \cdot d}{p} \quad (5.8)$$

Tendo estas duas distâncias torna-se possível calcular a distância  $r_1$ . Esta distância é a altura de dois triângulos rectângulos cujos catetos opostos aos ângulos  $\varphi_1$  e  $\varphi_2$  são  $d_1$  e  $d_2$ , respectivamente, o que torna possível calcular  $r_1$  através da relação trigonométrica que define a tangente de um ângulo. Como se dispõe de duas medidas para  $r_1$ , cada uma correspondendo a um par  $(d, \varphi)$ , o resultado final para esta distância é determinado através de uma média entre as duas medidas possíveis.

A distância  $r_2$  é calculada recorrendo ao ponto  $(x_c, y_c)$  previamente obtido e ao ponto médio dos dois pontos luminosos. Esta é calculada de forma semelhante às distâncias  $p_1$ ,  $p_2$  e  $p$ , podendo ser positiva ou negativa conforme o ponto médio dos dois *LEDs* se encontra à esquerda ou à direita da recta  $x=512$  previamente mencionada. O seu sinal será o mesmo do ângulo  $\theta$ , sendo a distância  $r$  calculada através do teorema de Pitágoras e dos dois catetos  $r_1$  e  $r_2$ .

Do ponto de vista de utilização passiva deste modo de funcionamento, a última função desempenhada pelo algoritmo de controlo é enviar as duas distâncias  $r_1$  e  $r_2$  para o programa de controlo do robô. Para utilizar este modo de funcionamento de uma forma activa foi criado um submodo denominado *follow-the-leader*. Neste submodo as distâncias  $r_1$  e  $r_2$  são utilizadas para controlar directamente as velocidades  $v$  e  $\omega$ , sendo o objectivo manter o robot, e por consequência o *Wiimote*, a uma distância constante da estrutura de *LEDs* e o ponto médio dos dois focos luminosos coincidente com a linha vertical central da imagem recebida pelo *Wiimote* ( $x=512$ ). O cálculo dos valores de actuação das velocidades angular e linear foi feito recorrendo ao sistema de inferência *Fuzzy* utilizado no modo volante, já que existem bastantes similaridades entres os dois modos de funcionamento, o que permite a utilização deste algoritmo sem alterações de raiz, visto que ambos os modos de funcionamento têm como objectivo final actuar sobre  $v$  e  $\omega$ , ambos têm duas entradas, acelerações nos eixos  $y$  e  $z$  no caso do volante e distâncias  $r_1$  e  $r_2$  neste modo, e em ambos os casos é difícil estabelecer limites entre

as várias zonas de actuação. Para utilizar o algoritmo *Fuzzy* previamente criado a única alteração necessária é estabelecer paralelismo entre as entradas  $r_1$  e a aceleração no eixo z e entre  $r_2$  e a aceleração no eixo y, sendo todo o restante sistema de inferência e cálculo igual ao utilizado no modo volante, com excepção do conjunto difuso positivo (P) da entrada  $r_1$  que teve de ser alterado para que a actuação sobre  $v$  fosse uniforme a partir de uma certa distância e até ao alcance máximo do sensor de infra-vermelhos do *Wiimote*, estando esta alteração retratada na figura 5.12.

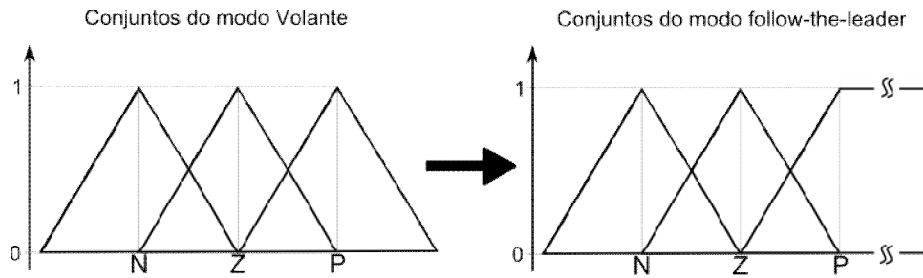


Figura 5.12 - Alteração dos conjuntos difusos da saída  $r_1$  no modo follow-the-leader

O modo *follow-the-leader* pode ser utilizado com os mesmos objectivos do modo volante, isto é, conduzir o robô até a um ponto, tendo a vantagem de ser mais fácil de utilizar e de se poder conduzir mais do que um robô em simultâneo, sendo, no entanto, necessário estabelecer distâncias de paragem diferentes para todos os robots para evitar colisões.

Na figura 5.13 está representado o fluxograma do modo de monitorização de infra-vermelhos.

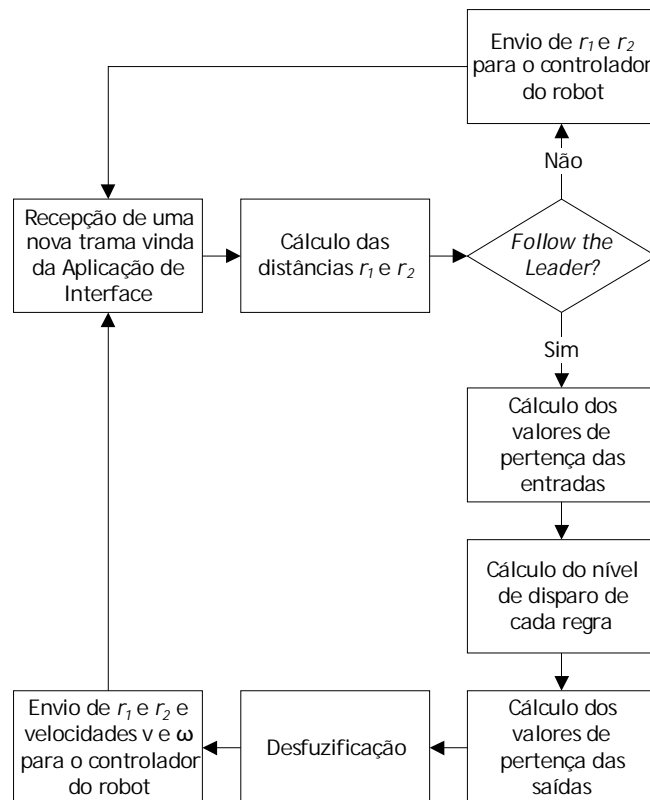


Figura 5.13 - Diagrama de fluxo do modo de monitorização de infra-vermelhos

### 5.3.3 Identificação de Movimentos

#### 5.3.3.1 Princípio de funcionamento

A capacidade mais inovadora introduzida pelo *Wiimote* foi a possibilidade de controlar um sistema através de movimentos. Esta capacidade já foi parcialmente explorada no modo volante, mas aí os acelerómetros eram utilizados para ler a aceleração da gravidade sendo necessário o *Wiimote* estar parado enquanto neste modo os acelerómetros serão utilizados para ler as acelerações criadas durante um movimento de forma a identificar esse movimento.

#### 5.3.3.2 Aplicação do algoritmo CBR

A análise das características dos movimentos já foi apresentada no capítulo dedicado à apresentação técnica do *Wiimote* (capítulo 3), sendo apenas necessário encontrar a técnica de controlo e análise que melhor se adapte a este modo de funcionamento. Esta técnica de controlo deve ser capaz de analisar as diferentes características que permitem diferenciar os diversos movimentos conhecidos, sendo também importante a capacidade de adaptação a novos tipos de movimentos e a diferentes utilizadores. Estas necessidades fizeram com que o estudo feito incidisse sobre as técnicas de controlo inteligente apresentadas no capítulo 2. Destas três técnicas de controlo inteligente a mais simples de implementar é o controlo difuso, no entanto esta técnica não foi utilizada devido ao volume de dados existente na entrada do sistema e à dificuldade de implementar uma base de regras capaz de o controlar eficazmente. Das duas técnicas restantes foi escolhido o controlo baseado em CBR devido à sua maior facilidade de implementação em relação às redes neuronais ou neuro-difusas e ao facto da complexidade exigida ao sistema não ser tal que tornasse a utilização de redes neuronais a melhor escolha. Outro factor que levou à escolha de raciocínio baseado em casos foi a capacidade de aprendizagem inata associada a este tipo de controlo, enquanto nos outros tipos de controlo inteligente esta capacidade tem de ser implementada através de algoritmos de adaptação.

No fluxograma da figura 5.14 está demonstrada a aplicação do algoritmo exposto no capítulo 2 referente aos sistemas CBR a este modo de funcionamento. Como em todos os outros modos de funcionamento o início dá-se após a recepção de uma trama válida vinda da aplicação de interface, mas ao contrário dos outros modos de funcionamento a actuação não é imediata, neste modo os valores das acelerações são guardadas e é feito um levantamento preliminar de certas características que irão possibilitar a identificação e cálculo dos atributos que definem o movimento quando este terminar. Estas características são o número de máximos e mínimos das acelerações parciais (acelerações segundo os eixos x, y e z), os seus valores e instantes em que ocorrem, sendo também calculado o módulo do vector aceleração a que o *Wiimote* está sujeito e a informação referente aos máximos e mínimos desta aceleração. Para garantir que o programa de controlo não responde a movimentos involuntários por parte do utilizador é necessário manter premido o botão B do *Wiimote* durante o movimento. Quando este botão é largado o movimento termina e se o número de amostras obtidas for suficiente dá-se início ao ciclo CBR.

O conjunto de vectores que contém dados sobre as acelerações lidas durante o movimento representa o novo caso e como foi apresentado anteriormente o primeiro passo do ciclo CBR é recolher informações sobre os atributos desse caso. Alguns atributos já foram recolhidos durante o processo de amostragem (informação referente a máximos e mínimos) sendo apenas necessário filtrá-los, outros necessitam de ser calculados através da informação contida nos

vectores de acelerações. O processo de cálculo e triagem de atributos será explicado à frente. Tendo todos os atributos identificados e calculados, o próximo passo do ciclo CBR é comparar o vector de atributos deste movimento com o vector de atributos de todos os movimentos guardados na base de casos, de forma a identificar o movimento mais semelhante ao que está a ser analisado. Para tal é calculado o somatório das diferenças quadradas entre cada atributo do movimento conhecido a ser analisado e do movimento a reconhecer, sendo a solução proposta pelo sistema de inferência CBR a descrição do movimento conhecido onde esse somatório é menor. Os dois últimos passos do ciclo CBR estão interligados e necessitam da presença humana para verificar se a solução proposta é a solução correcta. Caso seja, é necessário verificar se a diferença entre o novo caso e o caso mais próximo é suficientemente pequena de forma a garantir que no futuro movimentos semelhantes serão identificados correctamente, e se não for o novo caso deve ser adicionado à base de casos. Caso a resposta não seja a correcta, é necessário fornecer a resposta correcta e verificar qual a razão que levou o sistema a responder mal, podendo esta razão ser maus dados de entrada ou mau treino. Se se verificar que foi devido à primeira razão não é aconselhável que o novo caso seja adicionado à base de casos mas se a razão do erro for a segunda tal deve ser feito.

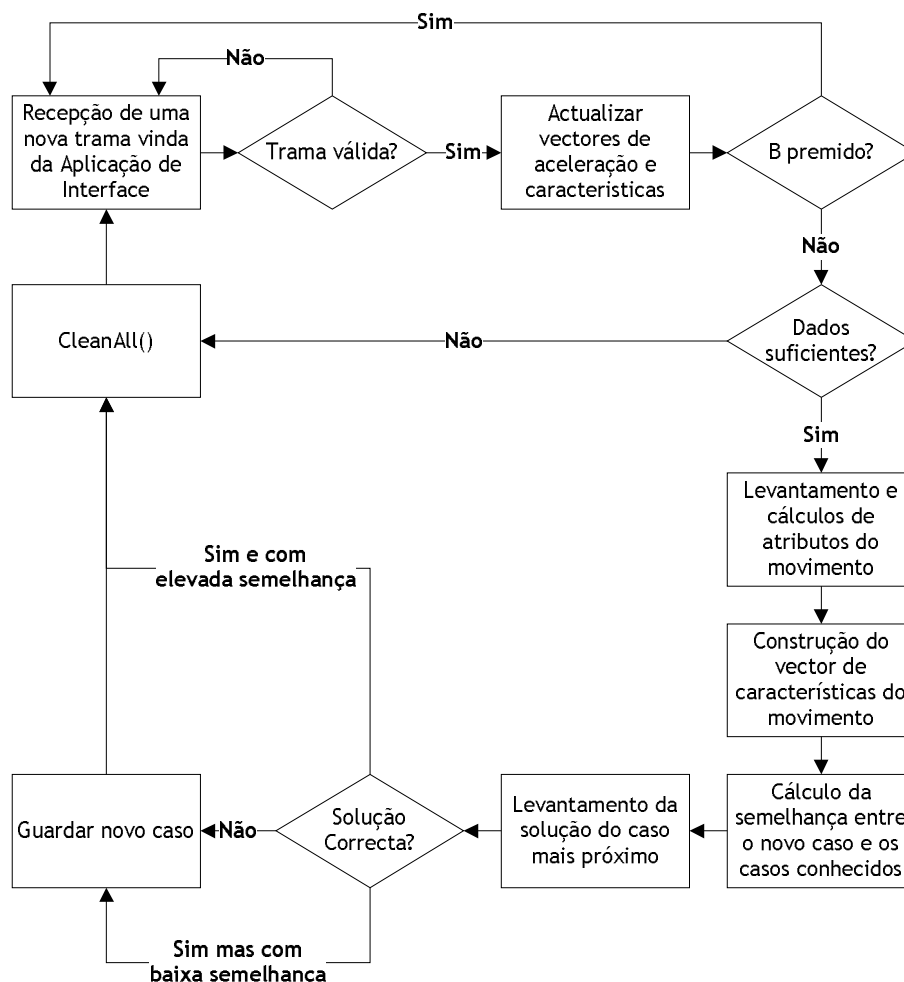


Figura 5.14 - Diagrama de fluxo do modo de identificação de movimentos

### 5.3.3.3 Detecção e Cálculo de Atributos

O processo de detecção de máximos e mínimos é feito aquando da recepção de uma nova trama vinda da aplicação de interface uma vez que o seu cálculo nesta fase do programa não atrasa a recepção da trama seguinte e reduz o tempo de processamento necessário após o final do movimento. Um ponto é considerado máximo se respeitar três condições: o valor da aceleração aumentar durante cinco amostras consecutivas, no mínimo, o próximo valor da aceleração é inferior ao valor actual e a diferença entre os valores da aceleração no instante de início da subida e no instante actual é superior a  $3.98 \text{ m/s}^2$  (valor determinado experimentalmente). Para ser possível fazer esta análise no momento da recepção de uma nova trama é necessário que ela seja feita com atraso de uma amostra, já que é necessário saber o valor da aceleração no instante seguinte devido à segunda condição apresentada, ou seja, a trama  $n$  só será analisada após ser recebida a trama  $n+1$ . As outras duas condições obrigam a que haja pelo menos uma amostra guardada antes de ser analisada a amostra  $n$ . A conjugação destas duas necessidades faz com que em  $k$  amostras apenas  $k-2$  sejam analisadas uma vez que não existem amostras anteriores à primeira nem amostras posteriores à última. A análise de mínimos é semelhante à análise de máximos sendo as três condições necessárias para um ponto ser mínimo as opostas às condições necessárias para ele ser máximo, isto é, o valor da aceleração diminui durante cinco amostras consecutivas, no mínimo, o próximo valor da aceleração é superior ao valor actual e a diferença entre os valores da aceleração no instante de início da descida e no instante actual é inferior a  $3.98 \text{ m/s}^2$  negativos ( $-3.98 \text{ m/s}^2$ ), o que implica que as restrições ao nível do número de amostras é igual às restrições impostas aos máximos.

Após esta análise preliminar é necessário filtrar os resultados para retirar máximos e mínimos locais que podem ser provocados por movimentos involuntários do utilizador ou por outras imperfeições do movimento, sendo esta filtragem feita em dois passos. No primeiro são rejeitados máximos cujo valor seja inferior à média da aceleração durante o movimento e mínimos cujo valor seja superior a essa média e no segundo passo é analisada a proximidade entre máximos e mínimos consecutivos. Se numa vizinhança em torno de um máximo existirem outros máximos então apenas será considerado máximo aquele que apresente maior valor e se numa vizinhança em torno de um mínimo existirem outros mínimos apenas será considerado como mínimo aquele que tiver o menor valor. A vizinhança foi determinada experimentalmente e vale dez amostras, aproximadamente 100ms.

Na figura 5.15 estão representados a aceleração lida num determinado eixo durante um movimento (a traço contínuo) e uma representação simplificada dessa aceleração que mantém as suas características principais (a traço descontínuo). Repare-se que os pontos mínimos nos instantes 0.29s, 0.4s, 0.53s e 0.69s não estão presentes na versão simplificada uma vez que estes são mínimos locais gerados durante o movimento e que não oferecem informação útil para a identificação do mesmo já que são desprezáveis perante os mínimos ocorridos nos instantes 0.21s, 0.45s e 0.62s. O mesmo acontece com os pontos máximos nos instantes 0.26s, 0.48s e 0.66s que são consequência dos mínimos locais já citados, sendo desprezados em relação aos máximos dos instantes 0.35s, 0.58s e 0.7s. A versão simplificada da aceleração é o resultado do algoritmo de eliminação de máximos e mínimos locais apresentado.



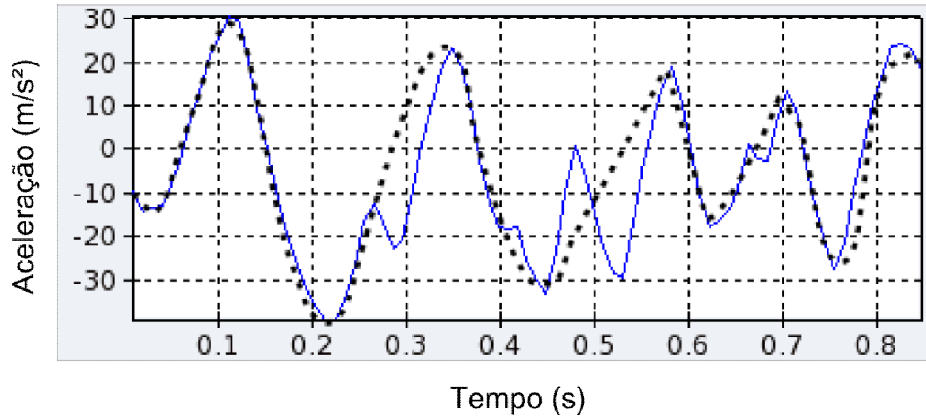


Figura 5.15 - Comparação da aceleração lida (a traço contínuo) com a aceleração obtida após o processamento do algoritmo de eliminação de máximos e mínimos locais (a tracejado),

Atendendo ao estudo feito anteriormente em relação às características das acelerações que permitem diferenciar os vários movimentos, após se ter obtido as informações referentes ao número de máximos e mínimos das acelerações e os seus valores apenas resta calcular as distâncias entre os pontos máximos e os pontos mínimos.

Após os primeiros testes do algoritmo chegou-se à conclusão que estes dados eram suficientes para identificar movimentos simples, como deslocamentos lineares num só sentido, mas não eram suficientes para distinguir movimentos mais complexos, como círculos ou movimentos repetitivos sobre um eixo. Foi então necessário calcular outros atributos que em conjunto com os já existentes permitissem identificar esses movimentos mais complexos. Após testes chegou-se à conclusão que a informação contida nos ângulos  $\alpha$  e  $\beta$ , representados na figura 5.16, e o valor da contribuição de cada uma das acelerações parciais para a aceleração total, isto é, o valor percentual dessa aceleração na composição da aceleração total, permitiam melhorar o desempenho do sistema ao ponto de ele diferenciar e identificar correctamente os diversos tipos de movimentos mais complexos. Estes dois ângulos podem ser vistos como indicadores do ponto tridimensional onde a aceleração se encontra num dado instante o que significa que através da análise do seu desenvolvimento ao longo do tempo se pode descobrir qual a trajectória desenvolvida. As contribuições são indicadores da composição da aceleração do *Wiimote* em cada instante e em conjunto com os ângulos reforçam a resposta correcta e aumentam a diferença para as outras possíveis respostas. Para calcular estes atributos começa-se por localizar os instantes de início e fim do movimento. De seguida divide-se o intervalo definido por esses dois instantes em cinquenta partes e calcula-se o valor dos dois ângulos e das contribuições em cada parte, através das equações 5.9 a 5.13, onde  $k$  é o instante de amostragem,  $accx_k$ ,  $accy_k$  e  $accz_k$  são as acelerações em cada um dos eixos cartesianos e  $acc_k$  é o módulo da aceleração nesse instante e  $avg_x$ ,  $avg_y$  e  $avg_z$  são os valores médios das acelerações durante o movimento.

$$\alpha_k = \tan^{-1} \frac{accy_k - avg_y}{accx_k - avg_x} \quad (5.9)$$

$$\beta_k = \tan^{-1} \frac{accz_k - avg_z}{\sqrt{(accy_k - avg_y)^2 + (accx_k - avg_x)^2}} \quad (5.10)$$

$$ctb_{x,k} = \frac{accx_k^2}{acc_k^2} \quad (5.11)$$

$$ctb_{y,k} = \frac{accy_k^2}{acc_k^2} \quad (5.12)$$

$$ctb_{z,k} = \frac{accz_k^2}{acc_k^2} \quad (5.13)$$

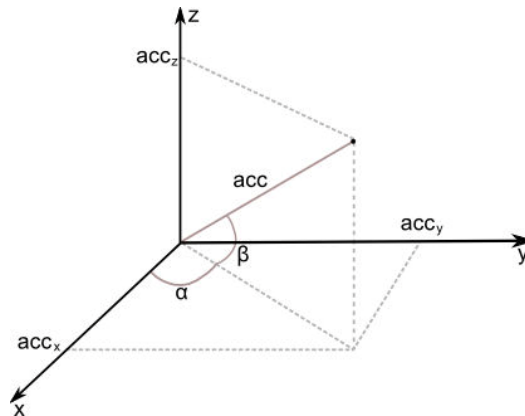


Figura 5.16 - Representação de  $\alpha$  e  $\beta$  no espaço tridimensional

#### 5.3.3.4 Criação da Base de Casos

Conhecendo todos os atributos necessários para identificar os movimentos resta apresentar a formatação da base de casos criada. Esta começa por identificar o número de movimentos nela guardada sendo cada um representado por trezentos e dezoito (318) atributos e por um identificador. A ordem por que estes estão guardados na base de casos é:

1. Identificador do movimento;
2. Número de máximos da aceleração no eixo x (no máximo cinco);
3. Número de mínimos da aceleração no eixo x (no máximo cinco);
4. Número de máximos da aceleração no eixo y (no máximo cinco);
5. Número de mínimos da aceleração no eixo y (no máximo cinco);
6. Número de máximos da aceleração no eixo z (no máximo cinco);
7. Número de mínimos da aceleração no eixo z (no máximo cinco);
8. Número de máximos do módulo da aceleração total (no máximo cinco);
9. Número de mínimos do módulo da aceleração total (no máximo cinco);
- 10 a 59. Valores da contribuição da aceleração segundo o eixo x;
- 60 a 109. Valores da contribuição da aceleração segundo o eixo y;
- 110 a 159. Valores da contribuição da aceleração segundo o eixo z;
- 160 a 209. Valores do ângulo  $\alpha$ ;
- 210 a 259. Valores do ângulo  $\beta$ ;
- 260 a 264. Distâncias entre máximos e mínimos da aceleração no eixo x;
- 265 a 269. Distâncias entre máximos e mínimos da aceleração no eixo y;

- 270 a 274. Distâncias entre máximos e mínimos da aceleração no eixo z;
- 275 a 279. Distâncias entre máximos e mínimos do módulo da aceleração total;
- 280 a 284. Valores dos máximos da aceleração no eixo x em relação à média;
- 285 a 289. Valores dos máximos da aceleração no eixo y em relação à média;
- 290 a 294. Valores dos máximos da aceleração no eixo z em relação à média;
- 295 a 299. Valores dos máximos do módulo da aceleração em relação à média;
- 300 a 304. Valores dos mínimos da aceleração no eixo x em relação à média;
- 305 a 309. Valores dos mínimos da aceleração no eixo y em relação à média;
- 310 a 314. Valores dos mínimos da aceleração no eixo z em relação à média;
- 315 a 319. Valores dos mínimos do módulo da aceleração em relação à média;

### 5.3.3.5 Funcionamento Offline

Para complementar este modo de funcionamento foi criada uma pequena aplicação que guarda os três vectores de acelerações lidas em ficheiros para tornar possível o seu uso futuro sem ser necessário o *Wiimote* estar conectado. A leitura destes ficheiros pode ser feita na aplicação de controlo através de uma pequena interface que lhe está adjacente. Desta forma torna-se possível guardar diversos movimentos feitos por vários utilizadores e testar o algoritmo CBR e a base de casos sem ser necessário possuir um *Wiimote* ou sem este estar conectado.

## 5.4 Actuação sobre o Robô

Tal como a comunicação entre as aplicações de controlo e interface, a comunicação entre a aplicação de controlo e o controlador do robô é feita através do envio de datagramas via *sockets* UDP. O datagrama enviado para o controlador do robô de limpeza é composto por uma *flag* booleana que indica se o utilizador pretende tomar controlo sobre o robô, um byte de comando e quatro valores reais. Este valores reais representam os valores de actuação sobre  $v$  e  $\omega$  calculados num dos controladores difusos e as distâncias  $r_1$  e  $r_2$  representadas na figura 5.7. O *byte* de comando contém informação sobre qual o modo de funcionamento que se encontra seleccionado pelo utilizador, estando esta informação gravada nos três *bits* menos significativos, e qual o movimento feito por este, caso o modo seleccionado seja o de identificação de movimentos, estando esta informação guardada nos cinco *bits* mais significativos, o que permite expandir o número de movimentos identificáveis até 32 ( $5bits - 2^5=32$ ). Se o modo seleccionado for o volante difuso ou o *follow-the-leader* o controlador do robô irá actuar sobre o este de acordo com os valores de  $v$  e  $\omega$  recebidos. Caso o modo seleccionado seja o monitor de infra-vermelhos o controlador do robô irá servir-se das duas distâncias enviadas no processamento dos algoritmos de auto-localização existentes. O modo de identificação de movimentos ainda necessita que sejam criadas as rotinas de médio/alto nível que irão representar as acções desempenhadas pelo robô associadas a cada movimento do utilizador.

## 5.5 Conclusão

Neste capítulo foram apresentados os detalhes da implementação das duas aplicações que constituem a interface homem-máquina desenvolvida. A Aplicação de Interface tem como

função garantir a comunicação com o *Wiimote* e analisar os relatórios periodicamente enviados por este. A Aplicação de Controlo analisa as informações úteis desses relatórios e usa-as num dos três modos de funcionamento implementados. No modo volante o utilizador pode utilizar o *Wiimote* como se se tratasse do volante de um automóvel e com isso controlar as velocidades linear e angular do robô. No modo de monitorização de infra-vermelhos o *Wiimote* é utilizado para medir as distâncias e orientações a que pares de fontes infra-vermelhas se encontram de si, podendo estas variáveis ser utilizadas em algoritmos de auto-localização ou na implementação de um modo de seguimento (*follow-the-leader*). No modo de identificação de movimentos é feita a identificação do movimento feito pelo utilizador, o que irá desencadear uma sequência de acções de médio/alto nível no robô associadas com o movimento feito. No desenvolvimento destes três modos foram implementados dois controladores inteligentes, um controlador difuso para controlar os modos volante e *follow-the-leader*, e um controlador de raciocínio baseado em casos, para controlar o modo de identificação de movimentos. Os seus tempos de processamento são 0.115ms e 8.712ms, respectivamente.

# Capítulo 6

## Análise de Resultados

Neste capítulo pretende-se analisar os resultados obtidos nos diferentes modos de funcionamento implementados.

### 6.1 Funcionamento como Volante

O primeiro modo de funcionamento implementado foi o volante difuso devido a ser aquele onde os fundamentos teóricos do seu princípio de funcionamento e da sua técnica de controlo são mais simples de implementar. Os resultados obtidos foram bastante satisfatórios uma vez que qualquer utilizador conseguia intuitivamente controlar o robot, no entanto era difícil, ou até impossível, mantê-lo imóvel, isto é, conseguir garantir que os valores de  $v$  e  $\omega$  calculados no sistema de inferência difuso quando o *Wiimote* era colocado na posição de actuação nula eram nulos. A solução encontrada para resolver este problema foi criar um *threshold* de 15% em redor do valor nulo de  $v$  e  $\omega$ , isto é, se o valor de saída do sistema difuso correspondente a estas variáveis fosse inferior a 0,15 o valor dessa variável que era enviado para o controlador do robot era o valor nulo. Após este ajuste passou a ser possível parar o robot mantendo o mesmo nível de precisão obtido anteriormente.

### 6.2 Monitorização de Infra-Vermelhos

O segundo modo de funcionamento implementado foi o monitor de infra-vermelhos. Para testar este modo de funcionamento foi construído um circuito muito simples constituído por dois emissores infra-vermelhos, duas resistências e uma bateria. A distância máxima a que o *Wiimote* conseguia detectar estas duas fontes luminosas era cerca de 2,5 metros mas apenas forma feitas medições até 2 metros devido à dificuldade de obter medições precisas a distâncias superiores. Para distâncias inferiores a 20cm não foi possível obter medições uma vez que o *Wiimote* não detectava as duas fontes tornando impossível utilizar o algoritmo de triangulação para calcular a distância a que estas se encontravam. No gráfico da figura 6.1 está representado

a variação do módulo do erro das medições em relação à posição real da estrutura de *LEDs*. Como se pode constatar o erro médio em módulo não ultrapassa os 3cm, tendo tendência para aumentar com o aumento da distância. Este aumento deve-se à perda de precisão do *Wiimote* a essas distâncias, já que grandes variações desta variável são representadas por pequenas variações das posições dos *pixéis* que representam os pontos luminosos, e à dificuldade de garantir que a estrutura de *LEDs* está colocada na posição correcta.

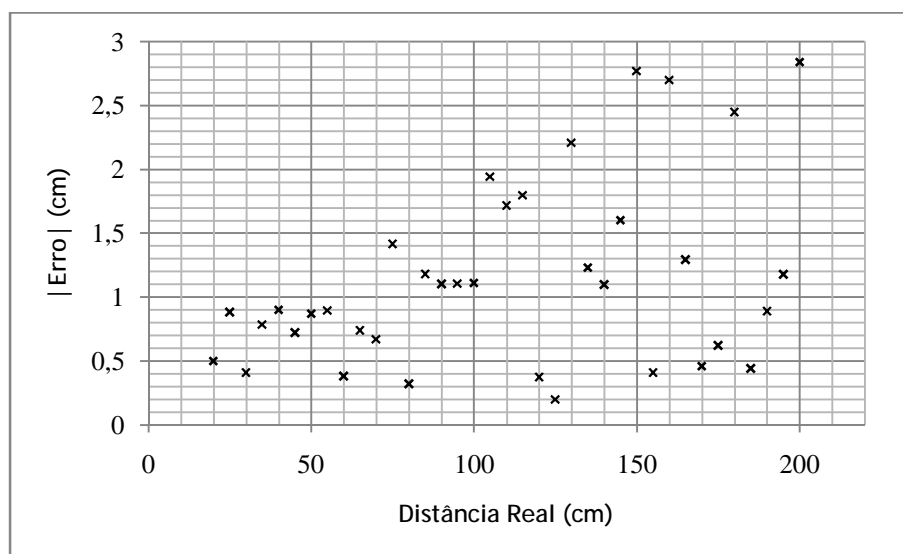


Figura 6.1 - Variação do módulo do erro das medições em relação à distância real

### 6.3 Identificação de Movimentos

Finalmente resta analisar os resultados obtidos no modo de identificação de movimentos. Foram analisados treze movimentos separados entre movimentos segundo cada um dos eixos e movimentos que englobavam diversos eixos. Os movimentos segundo um só eixo são: para a esquerda, para a direita e movimento repetitivo para a esquerda e direita, no caso do eixo x; para a frente, para a retaguarda e repetitivo nesses dois sentidos, no caso do eixo y; para cima, para baixo e repetitivo nesses sentidos, caso fosse o eixo z. Os movimentos que englobavam diversos eixos são os movimentos circulares e quadrados no plano xz, podendo estes ser horários e anti-horários. Terminada a fase de treino foi pedido a dez pessoas que fizessem os treze movimentos treinados de forma a ser possível avaliar a qualidade do algoritmo e da base de casos criada, estando os resultados expressos nos gráficos das figuras 6.2 a 6.5, podendo estes ser consultados em mais detalhe através das tabelas A2.2 a A2.5 no Anexo 2. É importante realçar que estas pessoas não tiveram qualquer participação na criação da base de casos.

Antes de analisar a qualidade das respostas obtidas é necessário relembrar que esta depende dos movimentos guardados na base de casos. Também é necessário ter em mente que a base de casos é composta por movimentos de uma só pessoa e que uma pessoa não realiza dois movimentos iguais, sendo as diferenças entre movimentos agravadas se se analisar os movimentos de pessoas diferentes.

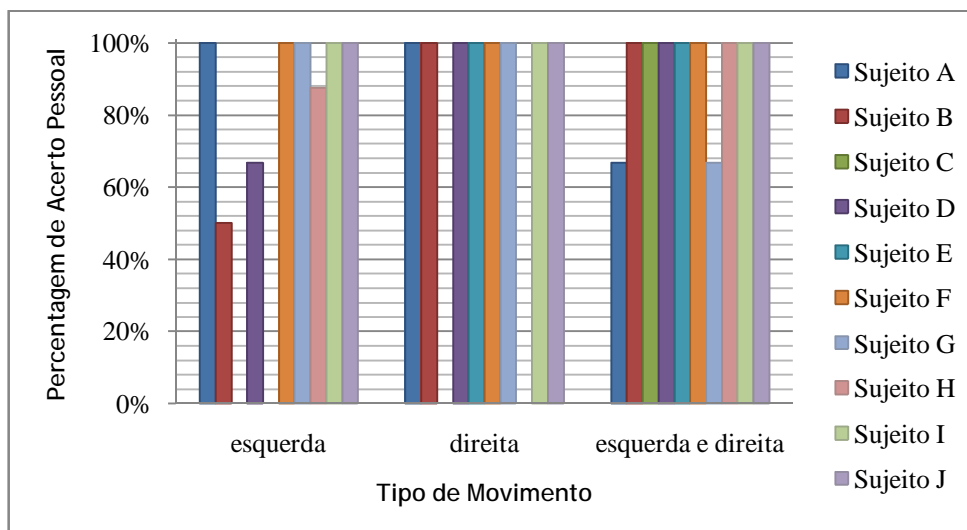


Figura 6.2 - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos ao longo do eixo x do Wiimote

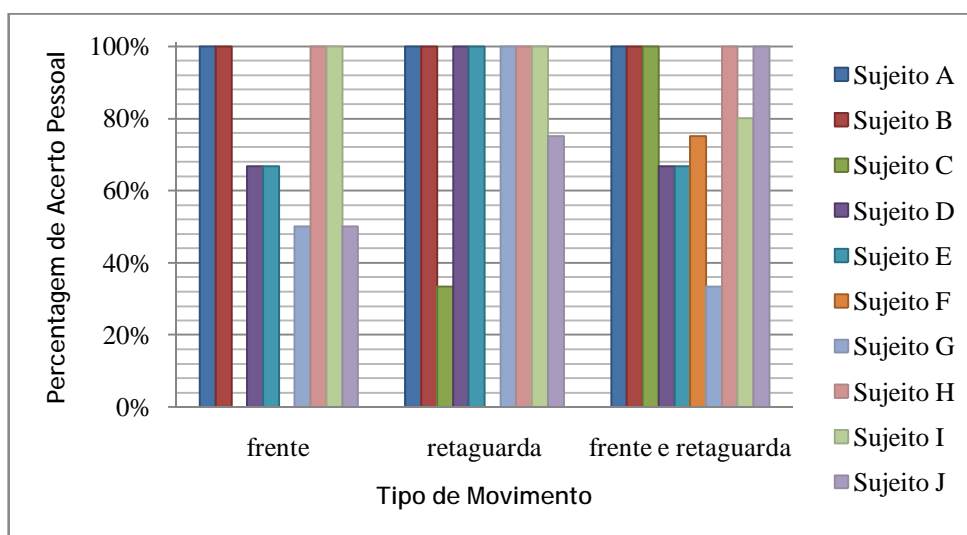


Figura 6.3 - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos ao longo do eixo y do Wiimote

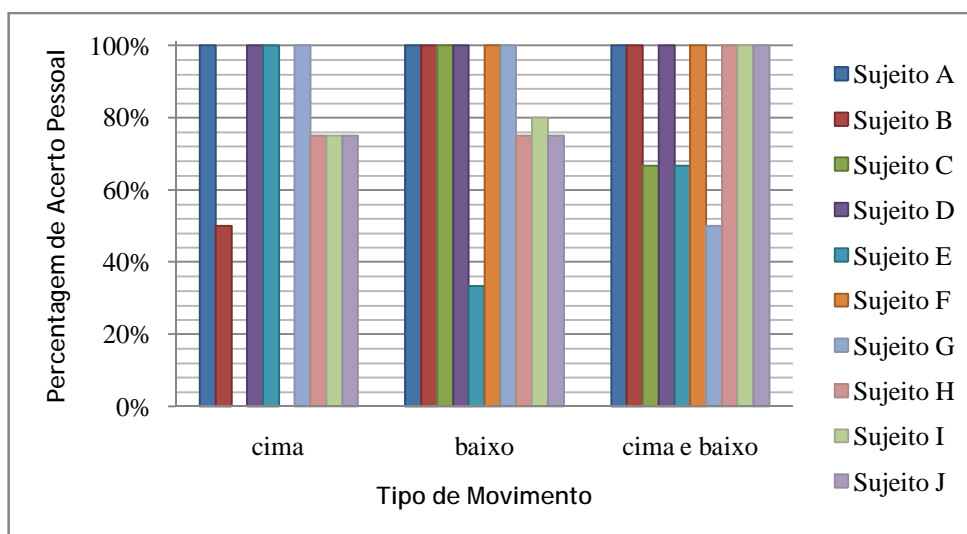


Figura 6.4 - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos ao longo do eixo z do Wiimote

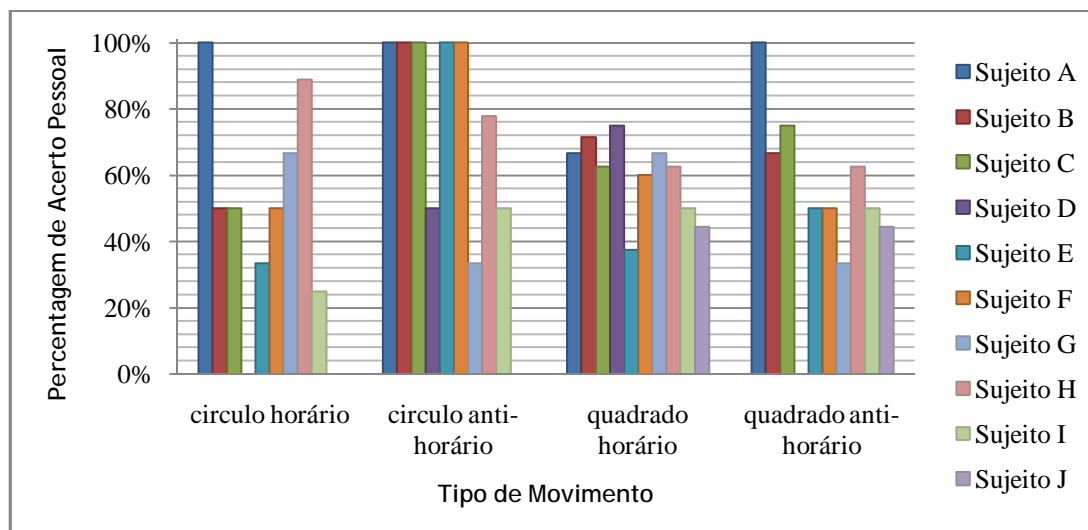


Figura 6.5 - Qualidade das respostas do algoritmo de identificação de movimentos para movimentos feitos no plano xz do Wiimote

Analisando as percentagens de acerto obtidas chega-se à conclusão que o algoritmo de análise genérica de um movimento introduzido no ciclo CBR é capaz de identificar correctamente a maioria dos movimentos das pessoas analisadas, com excepção de certos movimentos de alguns sujeitos por razões que serão explicadas à frente, sendo a taxa de sucesso deste conjunto de movimentos 67,71%. Estes resultados, apesar de inicialmente não o demonstrarem, são bons porque provam que, apesar dos movimentos guardados na base de casos serem de uma só pessoa, é possível identificar os treze movimentos de diversas pessoas. Outro indicador da boa qualidade das respostas obtidas é o facto de se conseguir diferenciar os sentidos dos movimentos mais simples, movimentos sobre um só eixo, e, sobretudo, o facto de se conseguir diferenciar círculos e quadrados e se estes foram horários ou anti-horários. Esta qualidade de resposta indicia que o algoritmo de diferenciação genérica introduzido no ciclo CBR funciona correctamente e que o sistema de indução CBR é um bom meio de detecção da resposta correcta. Para aumentar as percentagens de acerto seria necessário que o treino fosse feito por mais pessoas, o que iria introduzir maior variedade a cada um dos treze movimentos identificáveis e por consequência iria aumentar as taxas de sucesso de sistema.

A baixa capacidade de reconhecimento de alguns movimentos, tais como os movimentos segundo um eixo dos sujeitos C e F ou alguns movimentos mais complexos dos sujeitos D, E, G, I e J, prende-se, maioritariamente, com a má qualidade dos dados fornecidos ao sistema. Foi visto anteriormente que um movimento linear em qualquer sentido segundo um só eixo apresenta semelhanças com um período de uma onda sinusoidal, tendo sido essas semelhanças utilizadas para identificar os diversos atributos utilizados no algoritmo CBR. Se, por qualquer razão, os sinais recebidos pelo sistema não forem minimamente parecidos com esse período de uma sinusóide (se forem semelhantes a rampas, por exemplo), existe uma grande probabilidade do sistema de inferência ser induzido em erro e retornar uma resposta errada. Isto acontece quando o botão 'B', que indica o início do movimento quando pressionado e o fim deste quando largado, é pressionado após o "verdadeiro" início do movimento ou largado antes do seu "verdadeiro" fim. É este facto que leva a que 77,8% e 56,25% dos movimentos simples realizados pelos sujeitos C e F, respectivamente, não sejam correctamente identificados. Note-se que este efeito é atenuado ou inexistente em movimentos repetitivos ao longo de um eixo, o que gera taxas de sucesso superiores em relação aos movimentos lineares simples ao longo desse eixo. Outro factor que influencia a qualidade dos dados recebidos pelo sistema é a velocidade a



que o movimento é feito. Se o movimento for feito a baixa velocidade o SNR das acelerações lidas será igualmente baixo o que faz com que a qualidade da informação que entra no sistema, e por consequência a consequência dos atributos calculados para esse movimento, seja baixa, levando o sistema a responder incorrectamente. Foi esta a razão maioritária do insucesso no reconhecimento dos movimentos apresentados na tabela A2.5 referentes aos sujeitos D, E, G, I e J.

## 6.4 Conclusões

Os resultados obtidos quando o *Wiimote* é utilizado como volante difuso foram bastante satisfatórios, sendo no entanto necessária a criação de um *trigger* em torno do valor nulo de forma a facilitar a imobilização do robô.

Os resultados obtidos para a monitorização de infra-vermelhos e, consequentemente, para o modo *follow-the-leader* são igualmente satisfatórios uma vez que o erro absoluto máximo medido foi inferior a 3cm. Como medida de facilitar a utilização deste último modo garantiu-se que os valores de actuação sobre  $v$  e  $\omega$  se anulavam se não fosse detectada qualquer fonte infra-vermelha durante 3s.

O modo de identificação de movimentos também apresenta resultados satisfatórios, a taxa de sucesso na identificação de um movimento ronda os 70%, tendo em conta as restrições subjacentes ao controlo CBR. Estas restrições devem-se à subjectividade adjacente ao manuseamento do *Wiimote* que levam a que cada pessoa efectue um movimento de forma diferente das anteriores. Contudo através dos resultados obtidos prova-se que existem sempre algumas características comuns que permitem identificar os movimentos de várias pessoas, mesmo quando a base de casos é composta por movimentos de uma só pessoa. Também se prova que o raciocínio baseado em casos é uma forma viável de satisfazer os requisitos deste sistema de identificação quando são considerados os atributos enumerados nas páginas 32 a35. As razões que conduzem à incorrecta identificação de um movimento podem ser lacunas na base de casos, que por ser composta por movimentos de uma só pessoa pode não conter movimentos suficientemente similares ao novo caso, já que estes variam de pessoa para pessoa, ou má qualidade dos dados fornecidos ao sistema de inferência, podendo esta má qualidade se dever às acelerações recebidas não apresentarem parecenças mínimas com o esperado ou à baixa velocidade a que o movimento è feito o que gera baixas acelerações que em conjunto com o baixo valor do SNR conduzem à má identificação do movimento.



# Capítulo 7

## Conclusões e Trabalho Futuro

Terminado o trabalho prático e finalizada a análise de resultado resta apresentar as conclusões retiradas ao longo deste trabalho e a avaliação do sucesso em termos de cumprimento de objectivos e a avaliação de possíveis formas de melhoramento e implementação do sistema desenvolvido.

### 7.1 Satisfação de Objectivos

O objectivo principal deste trabalho era elaborar uma interface homem-máquina utilizando o *Wiimote*. Pretendia-se utilizar todos os elementos internos deste controlador remoto para implementar diversos modos de controlo e tal foi conseguido. Os modos de controlo implementados foram o volante difuso, o modo de monitorização de infra-vermelhos e o modo de identificação de movimentos, sendo nesses modos utilizados os acelerómetros em conjunto com o *Wiimote* parado, o receptor de infra-vermelhos em conjunto com uma estrutura de dois *LEDs* e os acelerómetros em conjunto com a estrutura de botões, respectivamente.

No primeiro modo o *Wiimote* é utilizado para controlar as velocidades linear e angular directamente tal como um volante e o par de pedais de aceleração e travagem de um automóvel, permitindo ao utilizador controlar a orientação, velocidade e sentido do movimento do robot directamente.

O modo de monitorização de infra-vermelhos implementado tem a possibilidade de ser utilizado passiva ou activamente. A forma passiva deste modo tem como finalidade fornecer os dados referentes à distância e orientação da estrutura de *LEDs* em relação ao *Wiimote* a certos algoritmos existentes no *software* de controlo do robot, tal como algoritmos de auto-localização, de forma a melhorar o desempenho destes algoritmos. A forma activa deste modo de funcionamento utiliza a distância e orientação calculadas para actuar sobre as velocidades angular e linear do robot de forma a manter o *Wiimote*, e por consequência o robot a que está acoplado, a uma distância fixa e orientado em relação à estrutura de *LEDs*. O erro máximo das medições obtidas neste algoritmo foi inferior a 3cm.

Finalmente, o modo de identificação de movimentos tem como objectivo final permitir ao utilizador controlar os movimentos do robot através dos seus movimentos. Foi estabelecido que

os movimentos a detectar seriam treze e dispersos pelos divididos em movimentos sobre um só eixo, apenas num sentido ou nos dois sentidos, e movimentos feitos em vários eixos, círculos e quadrados em sentido horário e anti-horário. Os resultados deste modo de funcionamento estão apresentados no capítulo 6 e revelam que apesar do algoritmo CBR utilizado responder com uma taxa de sucesso extremamente aceitável, tendo em conto que este foi treinado por apenas uma pessoa, a base de casos necessita de mais movimentos e/ou de movimentos realizados por outras pessoas de forma a aumentar a variedade de casos existentes e com isso aumentar o desempenho do algoritmo de identificação de movimentos. A qualidade da resposta do algoritmo também depende da velocidade a que o movimento é feito, uma vez que o valor de SNR é baixo e para baixas velocidades as acelerações lidas confundem-se com ruído, provocado pela vibração da mão por exemplo, o que leva o sistema de identificação a responder incorrectamente.

Do ponto de vista das técnicas de controlo utilizadas foram utilizadas duas técnicas de controlo inteligente, controlo baseado em lógica difusa nos modos volante e monitor de infra-vermelhos e raciocínio baseado em casos no modo de identificação de movimentos. O controlo baseado em lógica adequa-se aos dois primeiros modos devido ao baixo número variáveis envolvidas, o que torna simples a criação dos conjuntos difusos e da base de regras, e ao aspecto não-linear associado à análise das acelerações dos eixos  $y$  e  $z$  ou das distâncias  $r_1$  e  $r_2$ , existindo ainda a vantagem desta abordagem eliminar possíveis problemas nos limites entre zonas ou pontos de controlo. No modo de identificação de movimentos foi utilizado um algoritmo CBR visto que o volume de informação necessária neste modo de funcionamento inviabiliza a utilização de um algoritmo *Fuzzy* e a precisão de resposta desejada não é elevada ao ponto de tornar uma rede neuronal, que é bastante mais complexa que um algoritmo CBR, vantajosa. Funcionalmente este algoritmo utiliza a informação de atributos do movimento para identificar esse movimento por comparação com outros já reconhecidos previamente, sendo esses atributos os números de máximos e mínimos de cada uma das acelerações, tal como os seus valores nesses pontos e a distância entre eles, a sequência dos valores das posições angulares da aceleração total nos planos  $xy$  e  $yz$  (ângulos  $\alpha$  e  $\beta$ ) e a sequência de contribuições das três acelerações parciais para a aceleração total. Conclui-se que estes atributos permitiam identificar com precisão o tipo de movimento feito pelo utilizador.

O objectivo secundário foi parcialmente satisfeito, isto é, todo o *software* utilizado foi *freeware* no entanto a aplicação não é *cross-platform* devido à biblioteca *libcwiimote* que é exclusiva para Linux. A não satisfação deste objectivo veio em consequência do *trade-off* relativo ao uso desta biblioteca, já que das bibliotecas disponíveis esta era a mais estável e a que apresentava maior nível de desenvolvimento aquando do início do trabalho.

## 7.2 Trabalho Futuro

O trabalho desenvolvido satisfaz a maioria das necessidades da interface homem-máquina desejada, no entanto existe capacidade de evolução da mesma. Foi visto que o algoritmo de identificação de movimentos se encontra funcional e que a comunicação deste com o controlador do robô também se encontra implementada, necessitando ainda de ser implementado o algoritmo capaz de actuar sobre o robot de forma a este desempenhar a acção associada ao movimento feito pelo utilizador. Como já foi mostrado no capítulo 6 e na secção anterior é necessário expandir a base de casos existente de forma a abranger movimentos típicos

de diversas pessoas para que o desempenho do sistema de inferência CBR seja maximizado. Pode existir também a necessidade de implementar um novo sistema de inferência que analise os resultados do sistema CBR criado de forma a melhorar ainda mais o desempenho do sistema de detecção de movimentos. A necessidade deste sistema de inferência deve-se a certos movimentos gerarem conjuntos de respostas cujas proximidades para o caso a analisar são bastante semelhantes podendo o mais próximo ser a resposta errada. Poderia ser utilizado um sistema de inferência *Fuzzy* uma vez que é o mais simples e o volume de dados de saída do sistema CBR não o torna inviável.

O sistema de monitorização de infra-vermelhos poderia ser melhorado através da utilização de *PWM* o que traria vantagens na identificação de *LEDs* e consequente rejeição de fontes parasitas.

O nível de *feedback* dado ao utilizador poderá ser melhorado. No final deste trabalho apenas foram utilizados os quatro *LEDs* do *Wii mote* como meio de informar o utilizador de qual o modo activo, *LEDs* 1 a 3, e se o controlo manual do robô está activo, *LED* 4. A quantidade e qualidade de informação sobre o estado do controlo exercido pelo utilizador sobre o robô pode ser aumentada se for utilizado o *speaker* interno do *Wii mote* e a sua capacidade de vibração.



## Referências Bibliográficas

- [1] Informações genéricas da Nintendo Wii. Disponível em <http://en.wikipedia.org/wiki/Wii>. Acesso em 21/Junho/2008.
- [2] Informações técnicas sobre os modos de funcionamento do Wiimote e o formato dos relatórios a eles associados. Disponível em <http://wiibrew.org/w/index.php?title=Wiimote>. Acesso em 21/Junho/2008.
- [3] Lista de bibliotecas existentes para interface com o Wiimote (fim da página). Disponível em <http://www.wiili.org/index.php/Wiimote>. Acesso em 21/Junho/2008.
- [4] Apresentação dos projectos de Johnny Lee referentes à utilização do Wiimote. Disponível em <http://www.cs.cmu.edu/~johnny/projects/wii>. Acesso em 21/Junho/2008.
- [5] Apresentação da biblioteca CWiid. Disponível em <http://www.wiili.org/index.php/CWiid>. Acesso em 21/Junho/2008.
- [6] Utilização do Wiimote para controlar um robot de desminagem, o PackBot. Disponível em [http://news.sky.com/skynews/Home/Technology/Nintendo-Wii-Controller-To-Operate-Military-Robots/Article/20080341310765?lid=ARTICLE\\_1310765\\_Nintendo%20Wii%20Controller%20To%20Operate%20Military%20Robots&lpos=searchresults](http://news.sky.com/skynews/Home/Technology/Nintendo-Wii-Controller-To-Operate-Military-Robots/Article/20080341310765?lid=ARTICLE_1310765_Nintendo%20Wii%20Controller%20To%20Operate%20Military%20Robots&lpos=searchresults). Acesso em 06/Julho/2008.
- [7] Sítio oficial do PackBot. Disponível em <http://www.irobot.com/sp.cfm?pageid=171>. Acesso em 21/Junho/2008.
- [8] Braço robótico controlado pelo Wiimote. Disponível em <http://br.youtube.com/watch?v=QqK6FvNeSDE>. Acesso em 24/Junho/2008.
- [9] Controlo de um pequeno robot utilizando o Wiimote em conjunto com a biblioteca CWiid. Disponível em [http://br.youtube.com/watch?v=DnpSmtjdh\\_o](http://br.youtube.com/watch?v=DnpSmtjdh_o). Acesso em 24/Junho/2008.
- [10] Controlo de um manipulador robótico com o Wiimote. Disponível em <http://br.youtube.com/watch?v=0qEotHQgUsg>. Acesso em 24/Junho/2008.
- [11] Sítio oficial do projecto CleanRob. Disponível em <http://paginas.fe.up.pt/~cleanrob>. Acesso em 24/Junho/2008.

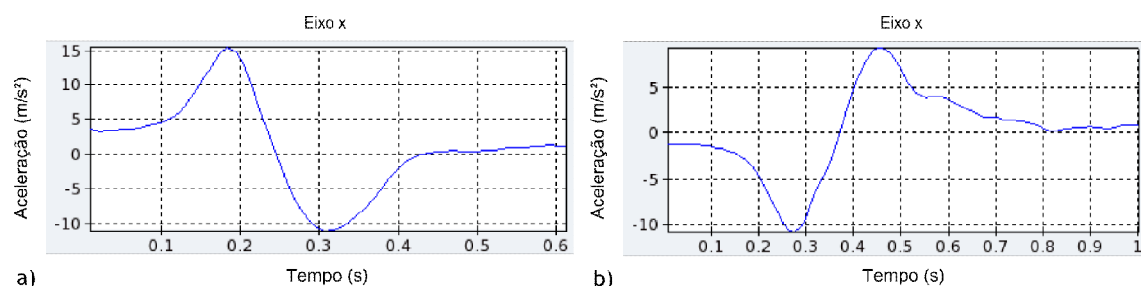
- [12] Informações técnicas dos sensores de medição de distâncias Sharp. Disponível em <http://www.active-robots.com/products/sensors/sharp/gp2y0a02yk.pdf>. Acesso a 24/Junho/2008.
- [13] G. Welch e G. Bishop, "An Introduction to the Kalman Filter ", *Technical Report 95-041 - Department of Computer Science University of North Carolina at Chapel Hill*, EUA, 2002
- [14] Ioannis M. Rekleitis, Gregory Dudek, and Evangelos Milios, "Probabilistic Cooperative Localization and Mapping in Practice", In *Proceedings of IEEE International Conference in Robotics and Automation*, 2003, <http://www.cim.mcgill.ca/~yiannis/particletutorial.pdf>.
- [15] O. Mayora-Ibarra, "From HCI to HRI - Usability Inspection in Multimodal Human-Robotics Interactions", *Proceedings of the 2003 IEEE International Workshop on Robot and Human Interactive Communication*, 2003.
- [16] Perzanowski, D.; Schultz, A.C.; Adams, W.; Marsh, E.; Bugajska, M. "Building a Multimodal Human-Robot Interface", *Intelligent Systems, IEEE*, vol. 16, Issue 1, Jan-Feb 2001 pp. 16 – 21.
- [17] H. Avilés, "Dynamic Bayesian Networks for Visual Recognition of Dynamic Gestures", *Journal of Intelligent & Fuzzy Systems*, vol. 12, pp. 243-250, 2002.
- [18] D. Driankov, *An Introduction to Fuzzy Control – Second Edition*, Berlin, Springer, 1996, capítulos 2,3 e 4.
- [19] J.-S. R. Jhang, *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*, Prentice Hall, 1997.
- [20] A. Aamodt and E. Plaza. "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" *Artificial Intelligence Communications* 7, no. 1, 1994, pp. 39-52.
- [21] J. Jantzen, "Design Of Fuzzy Controllers". Disponível em <http://www.iau.dtu.dk/~jjj/pubs/design.pdf>. Acesso em 24/Junho/2008.
- [22] Site oficial do KDevelop C++. Disponível em <http://www.kdevelop.org/>. Acesso em 21/Junho/2008.
- [23] Site da biblioteca libwiimote. Disponível em <http://libwiimote.sourceforge.net/>. Acesso em 21/Junho/2008.
- [24] Site oficial do protocolo Bluetooth oficial do Linux. Disponível em <http://www.bluez.org/>. Acesso em 21/Junho/2008.
- [25] Site oficial do ambiente de desenvolvimento Lazarus Free Pascal. Disponível em <http://www.lazarus.freepascal.org/>. Acesso em 21/Junho/2008.



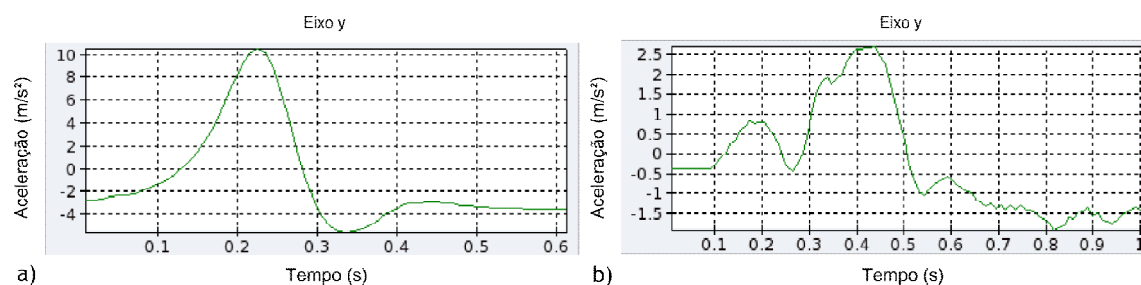
# Anexo 1

## Movimentos Analisados

Neste anexo são apresentados os gráficos típicos das acelerações obtidas em cada eixo durante os movimentos analisados e que levaram à posterior consideração dos atributos enumerados nas páginas 34 a 36. Para movimentos lineares as acelerações obtidas são, na grande maioria dos casos, bastante semelhantes às expostas de seguida, mas para os movimentos mais complexos, como círculos ou quadrados, essa semelhança pode-se resumir a apenas alguns dos atributos considerados. É importante também lembrar que o conjunto de gestos que definem como os movimentos são feitos, e por consequência as acelerações lidas, variam de pessoa para pessoa, podendo estes gráficos padrão não corresponder aos movimentos de certas pessoas.



**Figura A1.1** – Aceleração ao longo do eixo x para um movimento: a) linear para a esquerda, b) linear para a direita



**Figura A1.2** – Aceleração ao longo do eixo y para um movimento: a) linear para a esquerda, b) linear para a direita

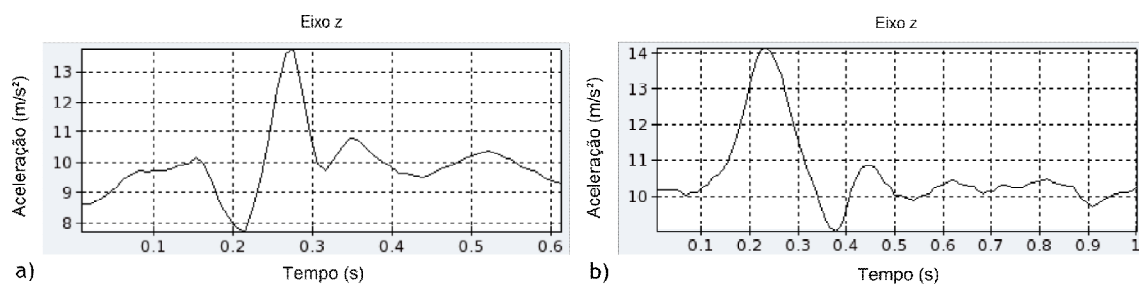


Figura A1.3 - Acelerações ao longo do eixo z para um movimento: a) linear para a esquerda, b) linear para a direita

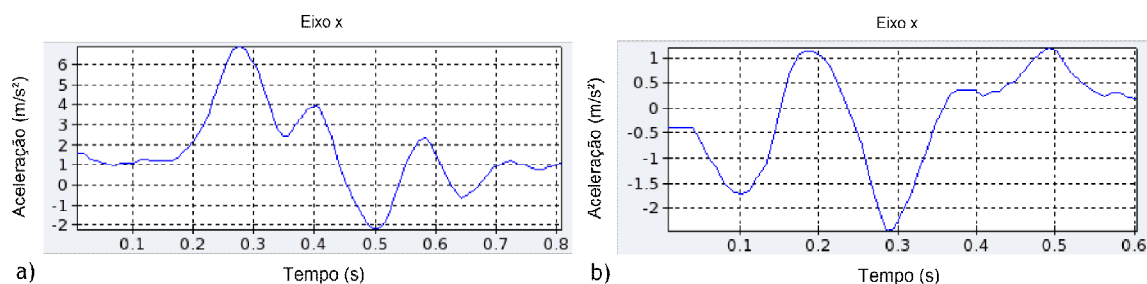


Figura A1.4 - Acelerações ao longo do eixo x para um movimento: a) linear para a frente, b) linear para a retaguarda

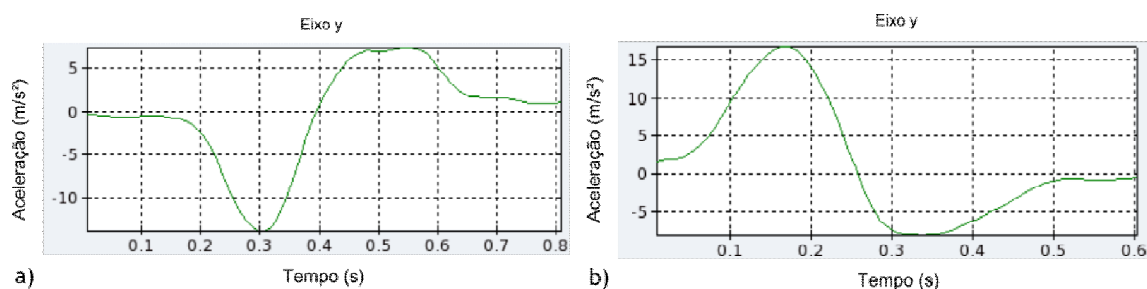


Figura A1.5 - Acelerações ao longo do eixo y para um movimento: a) linear para a frente, b) linear para a retaguarda

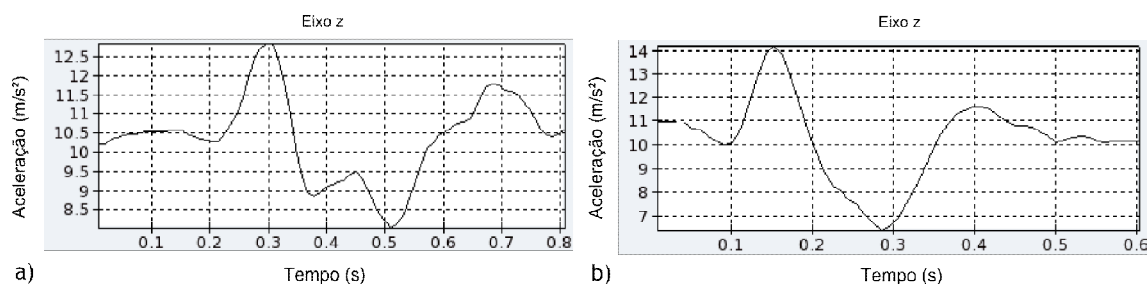


Figura A1.6 - Acelerações ao longo do eixo z para um movimento: a) linear para a frente, b) linear para a retaguarda

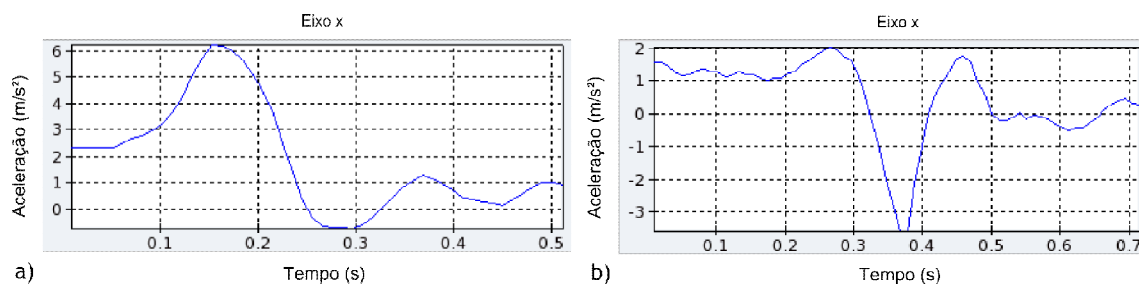


Figura A1.7 - Acelerações ao longo do eixo x para um movimento: a) linear para cima, b) linear para baixo

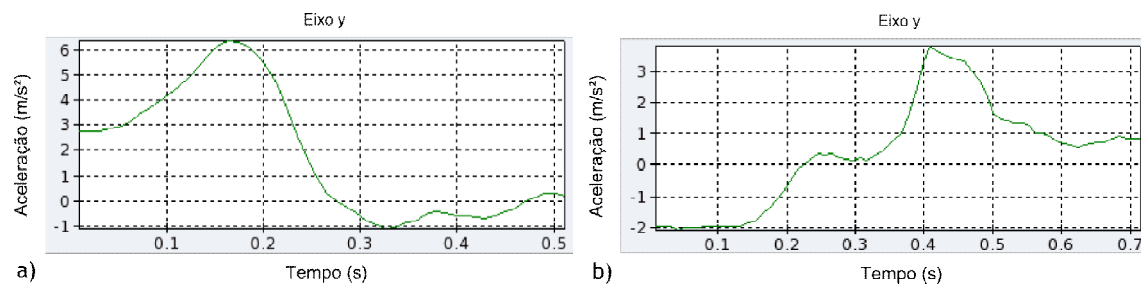


Figura A1.8 - Acelerações ao longo do eixo y para um movimento: a) linear para cima, b) linear para baixo

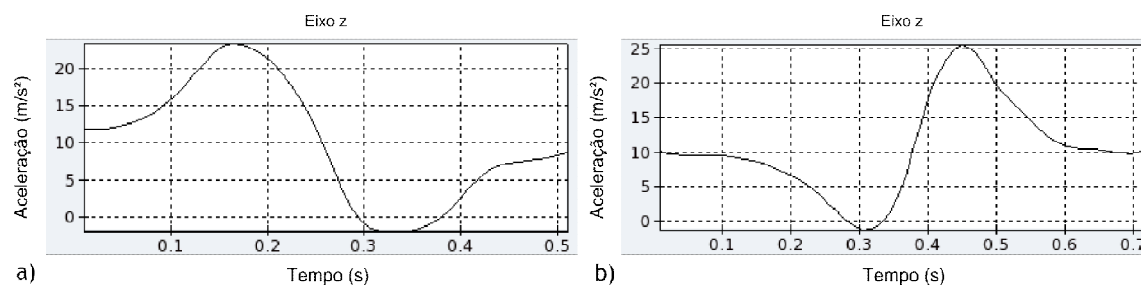


Figura A1.9 - Acelerações ao longo do eixo z para um movimento: a) linear para cima, b) linear para baixo

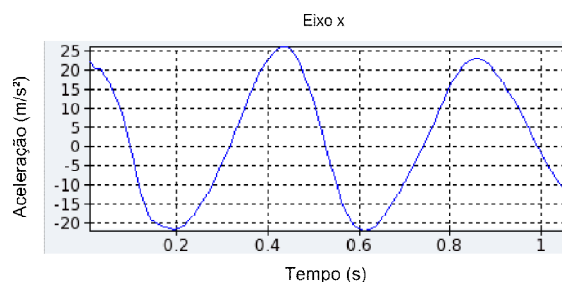


Figura A1.10 - Aceleração ao longo do eixo x num movimento linear repetitivo ao longo deste eixo

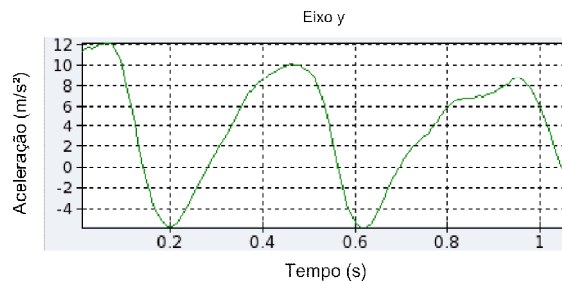


Figura A1.11 - Aceleração ao longo do eixo y num movimento linear repetitivo ao longo do eixo x

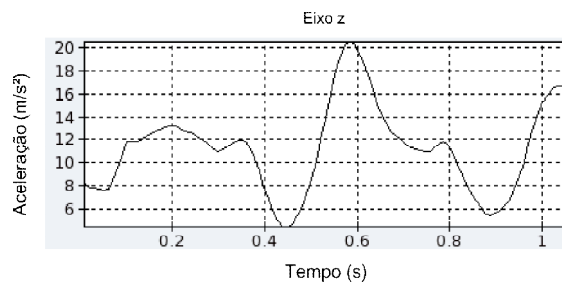


Figura A1.12 - Aceleração ao longo do eixo z num movimento linear repetitivo ao longo do eixo x

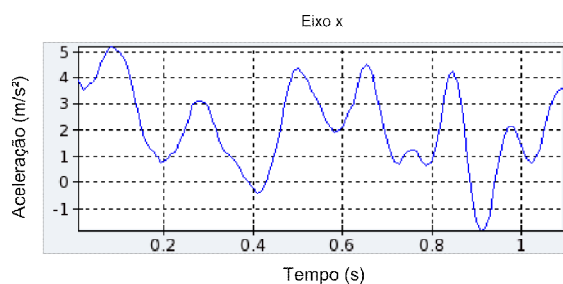


Figura A1.13 - Aceleração ao longo do eixo x num movimento linear repetitivo ao longo do eixo y

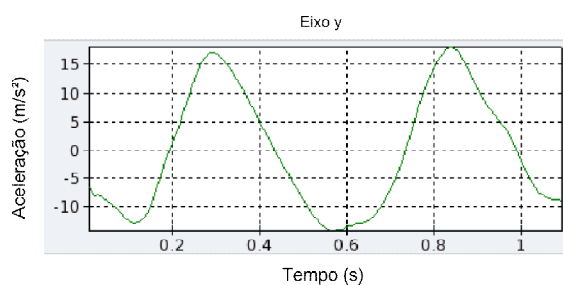


Figura A1.14 - Aceleração ao longo do eixo y num movimento linear repetitivo ao longo deste eixo

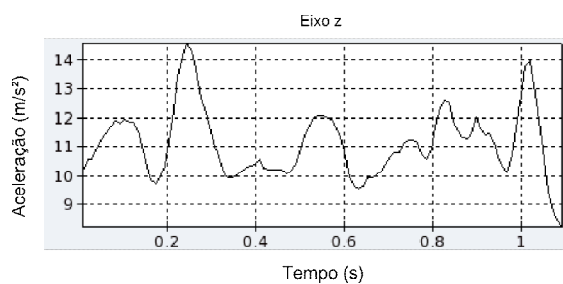


Figura A1.15 - Aceleração ao longo do eixo z num movimento linear repetitivo ao longo do eixo y

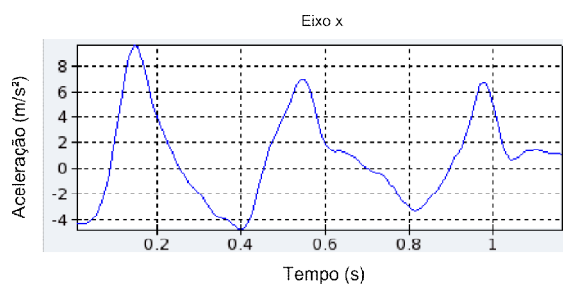


Figura A1.16 - Aceleração ao longo do eixo x num movimento linear repetitivo ao longo do eixo z

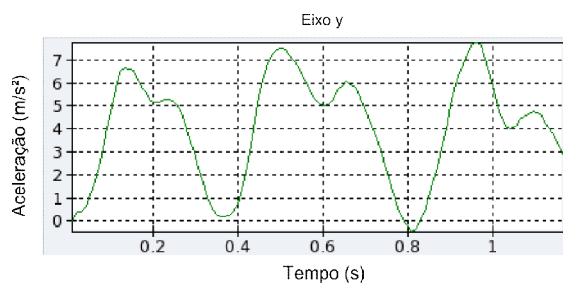


Figura A1.17 - Aceleração ao longo do eixo y num movimento linear repetitivo ao longo do eixo z

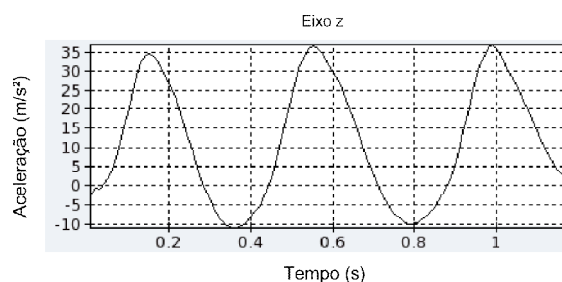


Figura A1.18 - Aceleração ao longo do eixo z num movimento linear repetitivo ao longo deste eixo

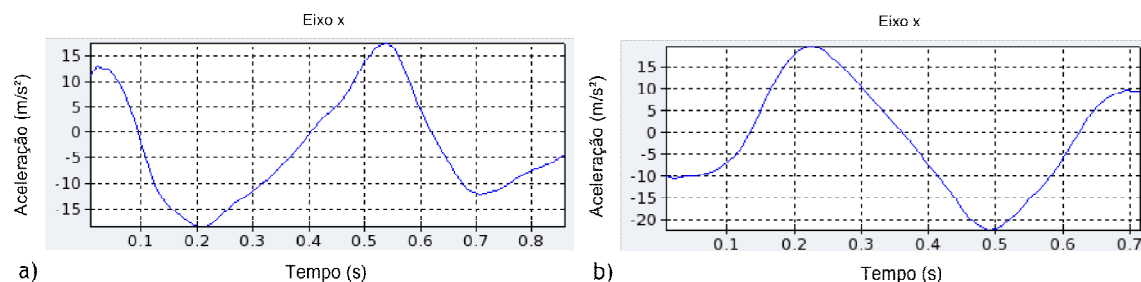


Figura A1.19 - Acelerações ao longo do eixo x para um movimento circular no plano xz no sentido: a) horário, b) anti-horário

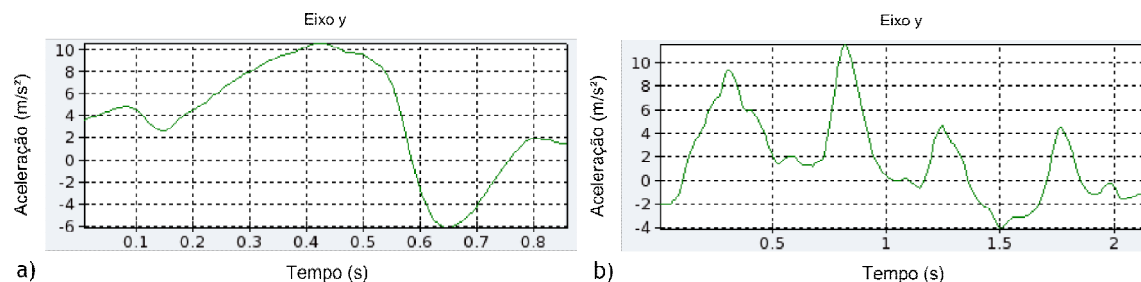


Figura A1.20 - Acelerações ao longo do eixo y para um movimento circular no plano xz no sentido: a) horário, b) anti-horário

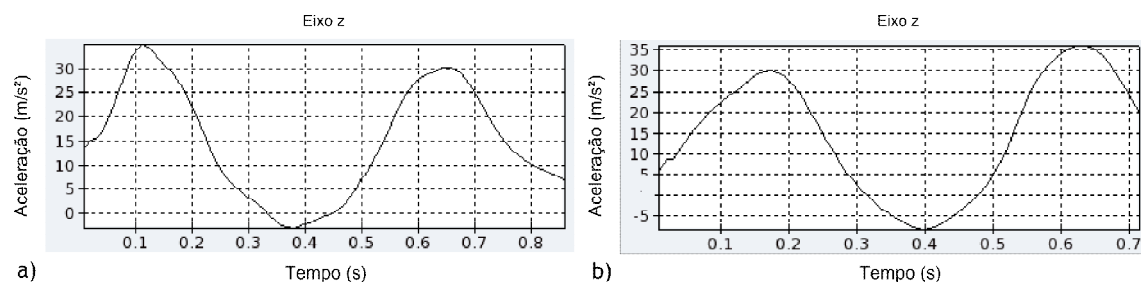


Figura A1.21 - Acelerações ao longo do eixo z para um movimento circular no plano xz no sentido: a) horário, b) anti-horário

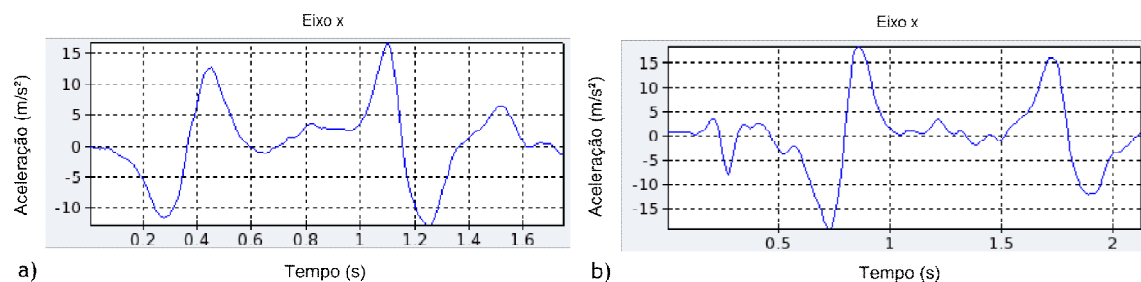


Figura A1.22 - Acelerações ao longo do eixo x para um movimento rectangular no plano xz no sentido: a) horário, b) anti-horário

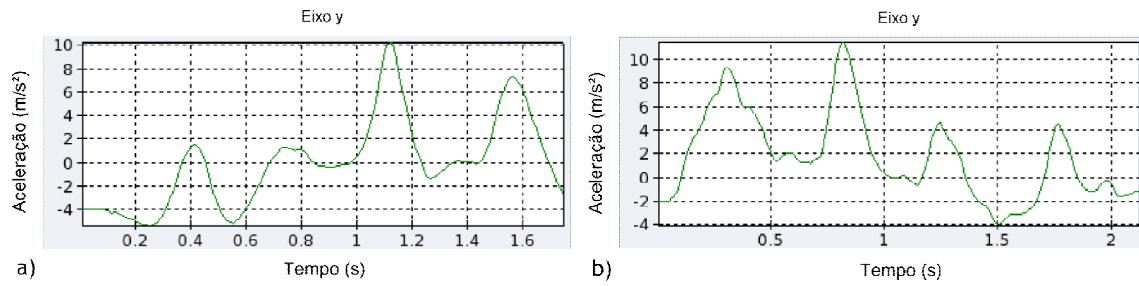


Figura A1.23 - Acelerações ao longo do eixo y para um movimento rectangular no plano xz no sentido: a) horário, b) anti-horário

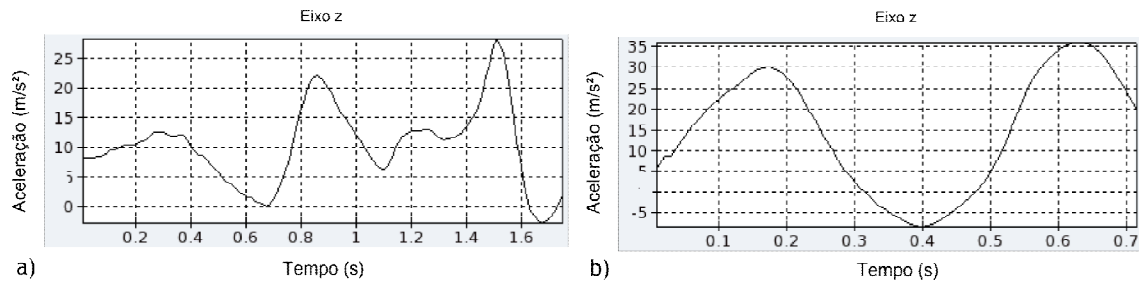


Figura A1.24 - Acelerações ao longo do eixo z para um movimento rectangular no plano xz no sentido: a) horário, b) anti-horário

## Anexo 2

### Tabelas de Resultados

Neste anexo são apresentadas as tabelas de resultados que permitiram chegar às conclusões expressas no capítulo 7. Na tabela A2.1 são apresentados os valores das distâncias entre a estrutura de *LEDs* infra-vermelhos e o *Wiimote* medidos com o receptor de infra-vermelhos em relação à distância real. Nas restantes tabelas são apresentadas as percentagens de acerto referentes ao algoritmo de identificação de movimentos.

Tabela A2.1 - Valores médios das distâncias medidas com o receptor de Infra-Vermelhos

| Real (cm) | Medida (cm) | Real (cm) | Medida (cm) | Real (cm) | Medida (cm) |
|-----------|-------------|-----------|-------------|-----------|-------------|
| 20        | 20,499      | 80        | 80,321      | 145       | 146,601     |
| 25        | 25,883      | 85        | 86,177      | 150       | 152,769     |
| 30        | 30,404      | 90        | 91,103      | 155       | 155,406     |
| 35        | 35,785      | 95        | 96,104      | 160       | 162,699     |
| 40        | 40,899      | 100       | 101,110     | 165       | 166,293     |
| 45        | 45,722      | 105       | 106,943     | 170       | 170,457     |
| 50        | 50,868      | 110       | 111,715     | 175       | 174,380     |
| 55        | 55,894      | 115       | 116,796     | 180       | 182,447     |
| 60        | 60,380      | 120       | 120,370     | 185       | 185,441     |
| 65        | 65,737      | 125       | 124,802     | 190       | 190,887     |
| 70        | 70,668      | 130       | 132,208     | 195       | 196,178     |
| 75        | 76,414      | 135       | 136,230     | 200       | 202,839     |
| 140       | 141,097     |           |             |           |             |

Tabela A2.2 - Resultados dos movimentos ao longo do eixo x

| Movimento<br>Sujeito |  | Esquerda | Direita | Esquerda<br>e Direita | Acerto pessoal |
|----------------------|--|----------|---------|-----------------------|----------------|
| A                    |  | 5/5      | 3/3     | 2/3                   | 90,91%         |
| B                    |  | 1/2      | 2/2     | 2/2                   | 83,33%         |
| C                    |  | 0/3      | 0/3     | 3/3                   | 33,33%         |
| D                    |  | 2/3      | 3/3     | 3/3                   | 88,89%         |
| E                    |  | 0/3      | 2/2     | 3/3                   | 62,50%         |
| F                    |  | 2/2      | 3/3     | 3/3                   | 100,00%        |
| G                    |  | 3/3      | 4/4     | 2/3                   | 90,00%         |
| H                    |  | 7/8      | 0/8     | 9/9                   | 64,00%         |
| I                    |  | 4/4      | 3/3     | 4/4                   | 100,00%        |
| J                    |  | 3/3      | 4/4     | 4/4                   | 100,00%        |
| Acerto de movimento  |  | 75,00%   | 68,57%  | 94,59%                |                |

Tabela A2.3 - Resultados dos movimentos ao longo do eixo y

| Movimento<br>Sujeito |  | Cima   | Baixo  | Cima<br>e Baixo | Acerto pessoal |
|----------------------|--|--------|--------|-----------------|----------------|
| A                    |  | 3/3    | 3/3    | 3/3             | 100,00%        |
| B                    |  | 1/2    | 3/3    | 3/3             | 87,50%         |
| C                    |  | 0/3    | 3/3    | 2/3             | 55,56%         |
| D                    |  | 3/3    | 2/2    | 2/2             | 100,00%        |
| E                    |  | 1/1    | 1/3    | 2/3             | 57,14%         |
| F                    |  | 0/3    | 2/2    | 3/3             | 62,50%         |
| G                    |  | 3/3    | 4/4    | 1/2             | 88,89%         |
| H                    |  | 6/8    | 6/8    | 7/7             | 82,61%         |
| I                    |  | 3/4    | 4/5    | 5/5             | 85,71%         |
| J                    |  | 3/4    | 3/4    | 4/4             | 83,33%         |
| Acerto de movimento  |  | 67,65% | 83,78% | 91,43%          |                |



Tabela A2.4 - Resultados dos movimentos segundo o eixo z

| Movimento<br>Sujeito |        |            |                     | Acerto pessoal |
|----------------------|--------|------------|---------------------|----------------|
|                      | Frente | Retaguarda | Frente e Retaguarda |                |
| A                    | 3/3    | 3/3        | 3/3                 | 100,00%        |
| B                    | 1/1    | 2/2        | 2/2                 | 100,00%        |
| C                    | 0/3    | 1/3        | 3/3                 | 44,44%         |
| D                    | 2/3    | 3/3        | 2/3                 | 77,78%         |
| E                    | 2/3    | 3/3        | 2/3                 | 77,78%         |
| F                    | 0/3    | 0/3        | 3/4                 | 30,00%         |
| G                    | 1/2    | 3/3        | 1/3                 | 62,50%         |
| H                    | 8/8    | 8/8        | 9/9                 | 100,00%        |
| I                    | 3/3    | 4/4        | 4/5                 | 91,67%         |
| J                    | 2/4    | 3/4        | 4/4                 | 75,00%         |
|                      |        |            |                     |                |
| Acerto de movimento  |        | 66,67%     | 83,33%              | 84,62%         |

Tabela A2.5 - Resultados dos movimentos feitos ao longo de vários eixos

| Movimento<br>Sujeito |                 |                      |                  |                       | Acerto pessoal |
|----------------------|-----------------|----------------------|------------------|-----------------------|----------------|
|                      | Círculo horário | Círculo anti-horário | Quadrado horário | Quadrado anti-horário |                |
| A                    | 4/4             | 3/3                  | 2/3              | 3/3                   | 92,31%         |
| B                    | 1/2             | 3/3                  | 5/7              | 4/6                   | 72,22%         |
| C                    | 1/2             | 3/3                  | 5/8              | 6/8                   | 71,43%         |
| D                    | 0/2             | 1/2                  | 6/8              | 0/0                   | 58,33%         |
| E                    | 1/3             | 2/2                  | 3/8              | 4/8                   | 47,62%         |
| F                    | 1/2             | 3/3                  | 3/5              | 2/4                   | 64,29%         |
| G                    | 2/3             | 1/3                  | 2/3              | 1/3                   | 50,00%         |
| H                    | 8/9             | 7/9                  | 5/8              | 5/8                   | 73,53%         |
| I                    | 1/4             | 2/4                  | 3/6              | 3/6                   | 45,00%         |
| J                    | 0/3             | 0/3                  | 4/9              | 4/9                   | 33,33%         |
|                      |                 |                      |                  |                       |                |
| Acerto de movimento  |                 | 55,88%               | 71,43%           | 58,46%                | 58%            |

